Optimisation de la consommation énergétique d'une ligne de métro automatique prenant en compte les aléas de trafic à l'aide d'outils d'intelligence artificielle

Jonathan Lesel

2016 - ENAM - 0018

École doctorale n° 432 : Sciences des Métiers de l'ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

École Nationale Supérieure d'Arts et Métiers

Spécialité "Génie électrique"

présentée et soutenue publiquement par

Jonathan LÉSEL

le 20 Juin 2016

Optimisation de la consommation énergétique d'une ligne de métro automatique prenant en compte les aléas de trafic à l'aide d'outils d'intelligence artificielle

Directeur de thèse : **Benoit ROBYNS**
Co-encadrants de thèse : **Patrick DEBAY - Gauthier CLAISSE**

Jury

M. Eric MONMASSON ,	Professeur, Laboratoire SATIE, Université de Cergy Pontoise	Président
M. Serge PIERFEDERICI,	Professeur, GREEN- ENSEM, Université de Lorraine	Rapporteur
M. Stéphane CAUX ,	Professeur, LAPLACE, Université de Toulouse	Rapporteur
M. Julien POUGET ,	Docteur-Ingénieur, SNCF I&R	Examinateur
M. Benoit ROBYNS ,	Professeur, L2EP-HEI	Examinateur
M. Patrick DEBAY ,	Docteur, HEI-Lille	Examinateur
M. Gauthier CLAISSE ,	Ingénieur, SIEMENS Mobility	Examinateur

Arts et Métiers ParisTech - Campus de Lille
EA 2697 - L2EP - Laboratoire d'Electrotechnique et d'Electronique de Puissance,F-59000 Lille, France

THÈSE

Table des matières

Table des figures

Liste des tableaux

Table des abréviations

ACO	Algorithme des colonies de fourmis ou *Ant Colony Optimization*
AG	Algorithme Génétique ou *Genetic Algorithm*
AR	Apprentissage par Renforcement
ATC	Automatic Train Control
CBTC	Communication Based Train Control
CCL	Coefficient de Corrélation Linéaire
CD	Coefficient de Détermination
CSPN	Coffrets de Surveillance du Potentiel Négatif
DC	Courant continu ou *Direct Current*
DNFQ	Dyna Neural Fitted Q-iteration
GIEC	Groupe Intergouvernemental d'Experts sur l'Evolution du Climat
HTA	Haute Tension niveau A ($1kV < U_n \leq 50kV$)
IA	Intelligence Artificielle
IEA	International Energy Agency
IEEE	Institute of Electrical and Electronics Engineers
IGBT	Insulated Gate Bipolar Transistor
MAE	Erreur Moyenne Absolue ou *Mean Absolute Error*
MC	Monte-Carlo (méthode d'optimisation)
MPE	Taux d'Erreur Moyen ou *Mean Percentage Error*
MSE	Marche Sur l'Erre
NFQ	Neural Fitted Q-iteration
NLE	Erreur Laplacienne Normalisée ou *Normalized Laplacian Error*
OEP	Optimisation par Essaims Particulaires ou *Paticle Swarm Optimization*
OEP-AG	Algorithme hybride d'optimisation OEP et AG
ONU	Organisation des Nations Unies
PA	Pilote Automatique
PBS	Propulsion and Braking System
PDM	Processus de Décision Markovien ou *Markovian Decision Process*
PEF	Poste Eclairage et Force
PESP	Periodic Event Scheduling Problem
PFD	Principe Fondamental de la Dynamique
PL	Programmation Linéaire
PLv	Poste de Livraison
PMC	Perceptron Multi-Couche ou *Multi-Layer Perceptron*
PR	Poste de Redressement
RBFR	Réseau à Base de Fonctions Radiales ou *Radial Base Function*
RMSE	Erreur Quadratique Moyenne ou *Root Mean Squarred Error*
RNA	Réseaux de Neurones Artificiels
RSE	Erreur Quadratique Relative ou *Relative Squarred Error*

RO	Recherche Opérationnelle
SARSA	*State - Action - Reward - State - Action* (méthode d'Apprentissage par Renforcement)
SLR	Systèmes Légers sur Rails ou *Light Rail Systems*
TCU	Unité de Contrôle de la Traction ou *Traction Control Unit*
TD	Méthode de Différences Temporelles ou *Temporal Difference*
TTP	Train Timetabling Problem
VAL	Véhicule Automatique Léger ou Villeneuve d'Ascq - Lille
VCU	Unité de Contrôle du Véhicule ou *Vehicle Control Unit*

Remerciements

Je tiens tout d'abord à remercier l'entreprise Siemens de m'avoir donner l'opportunité de travailler sur une thématique de thèse aussi innovante et intéressante.

Ensuite, mes remerciements vont à l'équipe encadrante : Benoit Robyns Directeur de la Recherche à HEI, Patrick Debay Enseignant-Chercheur à HEI, David Bourdon Chef de projet Siemens Mobility et Gauthier Claisse Ingénieur exploitation et maintenance Siemens Mobility. Je leur adresse toute ma gratitude pour leur disponibilité et les échanges fructueux qui m'ont permis de mener ce travail de recherche.

A ce titre, je remercie également l'ensemble des enseignants du L2EP ainsi que les équipes Siemens qui ont rendu ces trois années de travail riches et passionnantes autant d'un point de vue humain que professionnel.

J'exprime aussi mes remerciements aux membres du jury : Serge Pierfederici, Professeur à l'Université de Lorraine, Stéphane Caux, Professeur à l'INPT de Toulouse, Eric Monmasson, Professeur à l'Université de Cergy Pontoise et Julien Pouget, Chef de Projet à SNCF I&R.

Une pensée particulière pour les *compagnons de route* de la salle RR120 de HEI avec qui j'ai pu partagé des moments inoubliables. Anouar Bouallaga (dit *Le Magnifique*), Petronella Pankovits, le Talent venu d'Iran : Siyamak Sarabijeiranbolaghi, 我最好的汉语老师 Haibo Zhang, JC *Cheveux de Feu* Swierczek, Géraldine Ventoruzzo, Thang Do Minh, l'Indispensable Fabien Mollet, Riad Kadri, Faycal Bensmaine, Valentin Albinet, Pascal Monjean (mon premier gourou en LaTeX), Mouloud Guemri, Sid-Ali Amamra, Hayriye Gidik , Stojanka Petrusic, Lounes Koufi, Mohammed Amine Kenai,... et Laurent Taylor (Le corps libre et l'esprit apaisé... Big up à toi !).

Un remerciement chaleureux aussi pour deux collègues de Siemens avec qui j'ai partagé le même bureau pendant plus de deux ans : Bernard Boucher et Pierre-Yves Chantrel. Leurs débats enflammées resteront comme un de mes meilleurs souvenirs de thèse.

Une pensée pour les amis expatriés ou éparpillés en France, qui même s'ils n'ont pas pris part à ce travail de thèse, ont été une composante fondamentale de sa réalisation. D'après Julien Green, "*Le silence vaut mieux que n'importe quelle avalanche de paroles*", aussi les quelques lignes blanches suivantes leurs sont dédiées.

Je remercie également mes parents et mes sœurs, pour le soutien tout au long de mon cursus universitaire. Qui aurait cru que cela aboutirait à un doctorat ?!?! Vous avez toute ma gratitude et mon amour.

Un remerciement aussi pour la *belle-famille* lilloise qui a compensé le faible ensoleillement du Nord par une gentillesse et une joie de vivre qui ont rendu les conditions météorologiques de Lille plus supportable. Il ne sera pas dit que je serais ingrat avec la belle-famille du côté de Paris, une grosse pensée à vous aussi !!!

Enfin, ces dernières lignes sont dédiées à celle qui partage mon quotidien depuis plus de sept ans et qui est la source de mon bonheur : un grand merci à la douce Fiona pour son humour et sa patience...

Introduction générale

"*Désormais, la plus haute, la plus belle performance que devra réaliser l'Humanité sera de répondre à ses besoins vitaux avec les moyens les plus simples et les plus sains*". Cette citation de Pierre Rabhi illustre parfaitement les enjeux fondamentaux du XXI^e siècle ainsi que les défis à relever dans le domaine de l'optimisation pour assurer un développement durable.

Le phénomène d'urbanisation croissante qui a eu lieu au XX^e siècle et plus largement l'augmentation de la population mondiale a mis en exergue une problématique de réchauffement climatique, ainsi qu'un enjeu de raréfaction des sources d'énergie fossiles.

Dans ce contexte, la notion d'*efficacité énergétique* a vu le jour pour définir le besoin de diminuer la consommation de systèmes énergétiques tout en garantissant la fourniture de services identiques.

Appliqué aux transports urbains, ce concept impose de déterminer des solutions pour réduire le coût énergétique du transport des passagers.

La mise en œuvre de ce principe dans le cas de lignes de métro automatique revêt un intérêt tout particulier puisqu'à l'inverse d'autres modes de transports urbains supervisés par des conducteurs humains, dans un système totalement automatisé, il est possible de maitriser toutes les variables du système et le cas échéant de proposer des parades en cas de perturbations de trafic trop importantes.

Cette thèse a pour objectif d'utiliser des outils issus de l'intelligence artificielle afin de déterminer les règles de contrôle permettant de réaliser cet objectif d'efficacité énergétique, en agissant en temps réel sur le fonctionnement d'une ligne de métro pour gérer les aléas de trafic.

Ce rapport de thèse est scindé en cinq chapitres. Un premier chapitre est consacré d'une part à la présentation du contexte environnemental et sociétal qui a favorisé l'expansion des transports urbains, et d'autre part à une étude des travaux déjà menés sur la planification ferroviaire et sur l'optimisation énergétique de lignes ferroviaires.

Cet état de l'art permet d'identifier les axes d'études à suivre pour parvenir à réaliser les objectifs de cette thèse.

Le deuxième chapitre dresse tout d'abord un panorama des principaux éléments constitutifs d'une ligne de métro automatique avant de présenter une méthodologie de modélisation d'une ligne ferroviaire, puis d'effectuer une étude comparative de plusieurs méthodes de résolution itérative permettant de calculer les échanges de puissance s'effectuant entre les trains et les sous-stations d'alimentation.

Après avoir posé les bases du cahier des charges relatif à l'optimisation des paramètres d'exploitation, le troisième chapitre s'intéresse à l'optimisation de deux paramètres qui sont les plus influents sur l'évolution de la consommation énergétique d'une

ligne de métro automatique : l'intervalle d'exploitation et les temps de stationnement.

Ces travaux sont ensuite appliqués pour effectuer l'optimisation énergétique d'une ligne de métro sur une journée type dans un cas idéal d'exploitation.

Le quatrième chapitre de cette thèse a pour vocation de présenter une méthodologie pour rendre la démarche d'optimisation hors-ligne explicitée au troisième chapitre applicable en temps réel.

Pour ce faire, un estimateur neuronal est conçu pour synthétiser la méthode de résolution itérative. Cet estimateur a pour vocation de rendre la modélisation/résolution effectuée au chapitre 2 applicable en temps réel, en réduisant les temps de calcul de la boucle de résolution. Puis un apprentissage par renforcement est mis en œuvre pour déterminer une politique décisionnelle des temps de stationnement afin d'effectuer une optimisation énergétique dynamique prenant en compte les aléas d'exploitation.

Une étude des gains énergétiques potentiels générés par l'utilisation de cette méthode est ensuite accomplie.

Enfin, le dernier chapitre dresse un bilan des travaux effectués, des perspectives de développement pour améliorer la méthode et présente des exemples concrets d'optimisation énergétique de lignes ferroviaires.

Chapitre 1

Optimisation énergétique des tables horaires

« *They didn't know it was impossible so they did it.* »

Mark Twain

Sommaire

1.1 Introduction

Ce premier chapitre est composé de trois parties. Tout d'abord, le contexte sociétal et environnemental est présenté afin de situer les enjeux de l'étude. Cette partie introduit les problématiques liées à l'urbanisation croissante et à la raréfaction des ressources qui induisent le besoin de faire évoluer la gestion des transports ferroviaires urbains en augmentant leur efficacité énergétique.

Ensuite, le concept de la planification ferroviaire est défini afin d'expliciter la complexité de concevoir des tables horaires pour l'exploitant. Un état de l'art est réalisé pour décrire les différentes méthodes qui ont été utilisées au fil des décennies pour résoudre le problème de planification ferroviaire.

Enfin, quelques travaux proposant des solutions pour traiter le problème de l'optimisation énergétique de lignes ferrovaires sont analysés. Deux grands axes d'études sont décrits : la réduction des pics de puissance appelée et la réduction de la consommation énergétique globale des lignes ferrovaires.

Cette étude permet alors de souligner la nécessité de disposer d'un modèle énergétique précis permettant de calculer les flux de puissance qui se produisent lors des phases d'exploitation afin de prendre en compte le caractère non linéaire du freinage récupératif.

1.2 Contexte des travaux de thèse

1.2.1 Contexte historique

En 1662, sous l'impulsion de Blaise Pascal, la ville de Paris met en place le premier service de transport en commun du monde. Ce service composé initialement de 5 lignes, dont une permettait d'effectuer le tour de la périphérie de Paris, était assuré par des carrosses tractés par des chevaux.

Il faut ensuite attendre 1832 pour voir la première ligne de transport urbain à traction thermique être mise en service à New York. Dans les 50 années qui suivirent, la plupart des capitales européennes se sont dotées d'au moins une ligne de tramway similaire à celle de New-York.

Par la suite, l'urbanisation qui eut lieu au XXème siècle a vu une densification des réseaux de transports urbains, ainsi que l'abandon de la traction hippomobile au profit de la traction vapeur puis de la traction électrique.

C'est ainsi qu'en 1879, Werner Von Siemens fait la démonstration à Berlin de la première locomotive électrique sur un circuit circulaire, puis commercialise ce principe dès 1881 dans la périphérie de Berlin avec une ligne de tramway de 2,4km de long.

L'histoire des métros est quant à elle relativement récente : la première ligne de métro électrique voit le jour à Londres en 1890 et ce n'est qu'en 1983 que la première ligne de métro automatisée est ouverte à l'exploitation à Lille.

1.2.2 Contexte sociétal et environnemental

Ce sujet de thèse se trouve à l'intersection des trois piliers du développement durable, à savoir l'économie, l'écologie et le social. Il s'agit de faire évoluer les modes d'exploitation de lignes ferroviaires pour assurer à la fois une amélioration de leur efficacité énergétique, une réduction des coûts d'exploitation (ou tout du moins une stabilisation) et une maximisation de la satisfaction client.

1.2.2.1 Augmentation de l'urbanisation

Le XXème siècle a vu l'avènement de l'urbanisation, la population urbaine est passée de 15% à 45% au cours du siècle dernier. [1] et [2] estiment ainsi qu'en 2050, 70% de la population mondiale sera concentrée dans des villes. Cette urbanisation croissante s'accompagne également d'une augmentation des flux de mobilités urbains et périurbains.

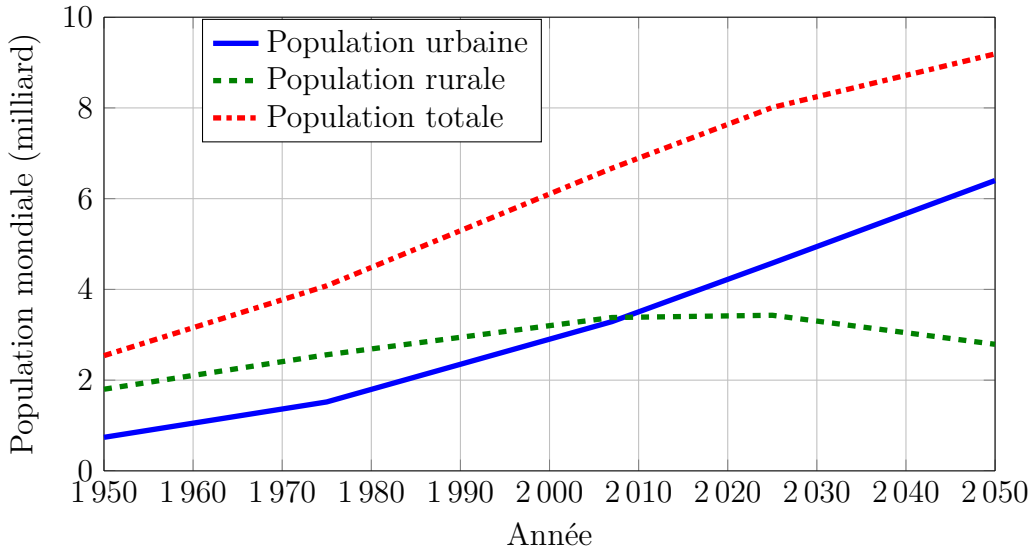


FIGURE 1.1 – Evolution de la population mondiale en valeur absolue.

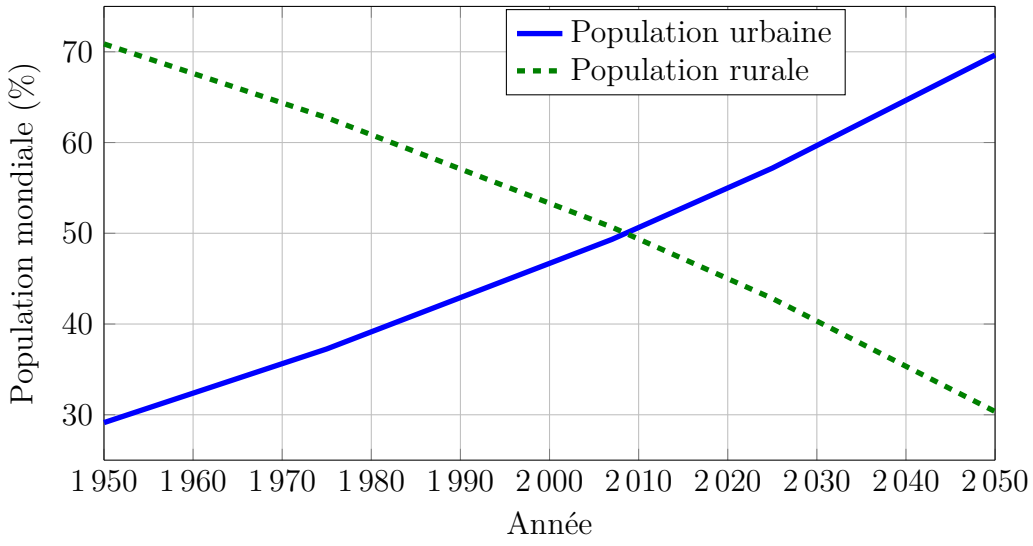


FIGURE 1.2 – Evolution de la population mondiale en valeur relative.

Les figures 1.1 et 1.2 sont issues des données fournies par l'ONU dans [3]. En 2050, les projections indiquent que la population mondiale va croître de 29% par rapport à 2010 pour s'établir à 9 milliards d'individus.

1.2.2.2 Épuisement des ressources fossiles et contraintes environnementales

Le développement démographique mondial s'accompagne également d'un accroissement de la consommation de ressources naturelles et notamment des énergies fossiles.

De 1973 à 2013, la consommation énergétique mondiale a ainsi plus que doublé et en 2012, 81,7% de cette énergie a été produite par la combustion d'énergies fossiles.

Ainsi, les scénarios les plus pessimistes estiment que les réserves en pétrole, gaz et matières fissiles seraient épuisées d'ici la fin du XXI^e siècle [4].

Le tableau 1.1 illustre la répartition de la consommation énergétique mondiale par secteur pour les années 1990 et 2012 [5].

Secteur	1973	1990	2012
Industrie	33%	29%	28%
Transport	23%	25%	27%
Résidentiel	-	24%	23%
Tertiaire	-	7%	8%
Agriculture	-	3%	2%
Autres	-	12%	12%

TABLEAU 1.1 – Répartition de la consommation énergétique mondiale par secteur.

D'après l'IEA [1], environ 40% de la consommation énergétique mondiale des transports est due aux déplacements urbains en 2012. Cependant, selon [2] la consommation des transports urbains est amenée à doubler d'ici 2050 du fait de l'augmentation des flux humains à l'intérieur des villes.

Outre la problématique de l'épuisement des ressources énergétiques fossiles, l'accroissement de la population mondiale a aussi un impact environnemental non négligeable. Depuis sa création en 1988, le GIEC [2] a multiplié les études et les rapports pour évaluer l'influence de l'activité humaine sur le réchauffement climatique et proposer des stratégies d'adaptation/atténuation.

Le GIEC a ainsi mis en lumière la nécessité de réduire les émissions de gaz à effets de serre afin de limiter le réchauffement climatique global [6], [7]. En 2014 les gouvernements européens ont adopté un nouveau plan d'action appelé *Plan Climat* dont l'objectif est de mettre en place une politique européenne énergétique à horizon 2030. Cette politique vise d'une part à réduire de 40% les émissions de gaz à effet de serre par rapport à celles de 1990, d'autre part à améliorer de 27% l'efficacité énergétique globale des infrastructures consommatrices et enfin de faire passer à 27% la part de la production énergétique issue de sources renouvelables.

1. International Energy Agency
2. Groupe Intergouvernemental d'Experts sur l'Evolution du Climat

1.2.2.3 Évolution des transports urbains

Dans ce contexte d'urbanisation croissante couplée au besoin accru de mobilité des hommes et des biens de consommation, l'évolution des transports urbains doit répondre à deux objectifs contradictoires.

Premièrement, la nécessité d'augmenter l'offre de transport afin de faciliter les flux humains et matériels entre les tissus urbains et péri-urbains. Deuxièmement, l'adoption d'une démarche environnementale responsable pour limiter l'impact écologique de la mondialisation.

Depuis quelques années, on assiste au niveau des villes, à un réaménagement des réseaux de transports visant à privilégier le développement des transports collectifs électriques, comme les tramways, les bus électriques ou encore les métros, au détriment des systèmes thermiques plus polluants.

Cependant comme le montre la figure 1.3, appelée *hyperbole de Newman-Kenworthy*, il existe une forte corrélation entre la densité urbaine et l'énergie dépensée pour le transport des personnes [8]. Les villes des pays développés les moins denses sont celles qui consomment le plus de pétrole pour les transports de personnes.

Bien que cette courbe se base sur des données de 1989, elle permet d'entrevoir les perspectives d'une utilisation plus massive des transports en commun et de l'installation de réseaux ferroviaires urbains électriques.

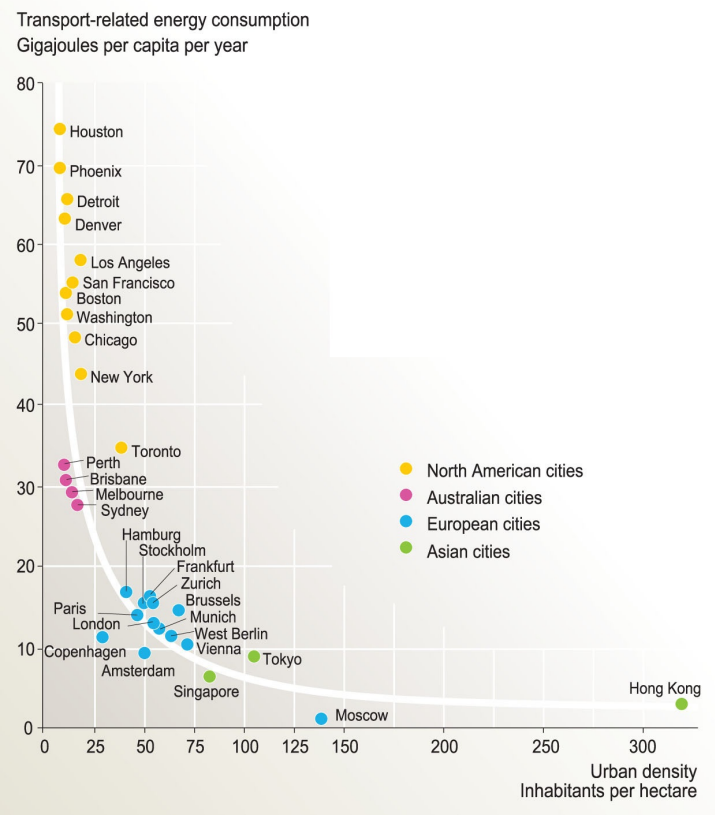


FIGURE 1.3 – Relation entre la consommation énergétique annuelle des transports par habitant et la densité urbaine.

A l'échelle d'entreprises comme Siemens, qui développent des systèmes de transports urbains électriques, les directives européennes et les prises de conscience écologique se

traduisent par le besoin de développer des systèmes toujours plus performants en terme de consommation énergétique tout en assurant la même qualité de service.

Ce dernier point est important, puisque du fait de l'augmentation démographique, l'optimisation énergétique ne doit pas entraîner de diminution des flux de passagers en transit.

1.2.2.4 Solutions envisagées

Les moyens pour réduire la consommation énergétique de lignes de métro sont principalement de trois types : la réduction des performances du matériel roulant, l'utilisation de profils d'éco-conduite et la récupération de l'énergie cinétique issue du freinage.

La réduction des performances du matériel roulant n'est pas une solution envisagée dans cette étude puisque cela nécessiterait d'exploiter plus de trains pour assurer la même qualité de service.

Elle se caractérise notamment par la limitation du taux d'accélération et de la vitesse commerciale des trains pour réduire la consommation globale de la ligne.

L'établissement de profils d'éco-conduite repose sur l'exploitation de la topographie de la ligne pour réduire la consommation énergétique des trains, principalement en insérant des phases de marche sur l'erre sur certaines interstations afin de tirer avantage de la déclivité de la voie.

La marche sur l'erre (MSE) consiste à allonger le temps de parcours des trains en coupant l'alimentation de leurs moteurs sur des portions d'interstation.

Les trains utilisent l'inertie acquise lors de l'accélération pour diminuer la consommation énergétique sur le reste de l'interstation, en ne consommant pas d'énergie de traction lors de ces phases.

Cette approche a été utilisée par [9] pour déterminer l'insertion optimale de phases de marche sur l'erre.

[10] a également étudié l'impact des techniques d'éco-conduite sur la consommation en définissant des profils de vitesse caractérisés par des hautes performances d'accélération/décélération et l'utilisation de phases de marche sur l'erre.

Bien que cette démarche semble prometteuse puisque [9] et [10] ont tous les deux enregistré une diminution de la consommation d'énergie de quelques dizaines de pour cent, une dégradation des temps de parcours a été constatée dans le même temps.

Ainsi, cette approche n'est pas étudiée ici puisqu'elle engendre une diminution de la qualité de service pour un même nombre de trains en ligne.

Dans une ligne de métro classique, le matériel roulant est équipé de deux systèmes de freinage : un frein électrique et un frein mécanique.

La récupération de l'énergie cinétique d'un train se fait donc en maximisant l'utilisation du frein électrique pour transformer cette énergie cinétique en énergie électrique réutilisable par la suite.

L'énergie électrique générée peut alors soit être stockée dans un système de stockage énergétique fixe ou embarqué, soit être renvoyée sur le réseau HTA (Haute Tension niveau A : $1kV < U_n \leq 50kV$) à l'aide de sous-stations réversibles ou être réutilisée par les autres trains circulant sur la ligne.

C'est cette dernière solution qui est étudiée dans ces travaux de thèse puisqu'elle permet d'effectuer l'optimisation énergétique d'une ligne déjà existante sans nécessiter de gros investissements matériels.

1.3 Planification ferroviaire

Avant de rentrer dans les détails de l'optimisation énergétique et de présenter les différentes études qui ont servi de point de départ à ces travaux, il convient de proposer un aperçu de la planification ferroviaire et de ses spécificités.

La planification ferroviaire s'effectue en deux étapes : une planification stratégique et une planification opérationnelle.

La planification stratégique dite planification réseau a pour but de définir l'infrastructure du réseau, tandis que la planification opérationnelle a pour objectif d'établir les horaires de passages des trains pour assurer le transit des passagers et également d'assigner un équipage à chaque train en exploitation [3].

1.3.1 Planification stratégique et Planification opérationnelle

La planification stratégique est effectuée sur un horizon long terme pour dimensionner les infrastructures du réseau ferroviaire et, dans des cas de lignes complexes, prévoir les possibilités d'interconnection entre les lignes pour faciliter le transit des passagers en limitant leur temps d'attente.

Pour cela, l'exploitant a besoin d'effectuer une estimation des flux de passagers sur les différentes lignes, de prévoir des routes ou des voies d'accès pour faciliter les flux humains et enfin de choisir le tracé des lignes du réseau ferroviaire.

La planification opérationnelle est effectuée sur un horizon moyen terme. Elle consiste à prévoir le plan de charge des véhicules et du personnel de sorte à minimiser les besoins matériels et humains pour assurer le service des tables horaires.

Il est à noter que certains auteurs considèrent la conception de tables horaires comme étant une étape de la planification stratégique tandis que d'autres considèrent qu'elle est une étape faisant l'interface entre la planification stratégique et la planification opérationnelle [11].

1.3.2 Évaluation de la capacité

La capacité d'une ligne peut se définir comme le nombre maximal de trains pouvant circuler dans un intervalle de temps donné dans des conditions d'exploitation réalistes et pour une structure de ligne, une structure d'horaire et une qualité de service données [12].

En outre, le problème de l'évaluation de la capacité permet de concevoir une offre répondant à une demande future probable sans surdimensionner les infrastructures de la ligne. Il s'agit ainsi de l'une des approches les plus simples pour concevoir des tables horaires performantes.

L'évaluation de la capacité d'une ligne fait intervenir de nombreux paramètres tels que :

— **les infrastructures de la ligne** : nombre de voies, signalisation, vitesse autorisée, maintenance, cisaillements de voies, ...

3. Parallèlement, au transit des passagers, cette étape permet également de gérer le transit du fret sur les lignes.

— **le plan de transport** : l'ordonnancement des trains, les contraintes horaires dues aux correspondances ou aux dessertes de voyageurs, ...

— **la qualité de service** : elle est une mesure du service fourni aux usagers de la ligne selon différents critères comme le nombre de passagers transportés par heure ou la fréquence des trains, le confort ainsi que les informations fournies aux usagers, ...

Dans ces travaux, la qualité de service fait référence à la fréquence nominale de passage des trains dans chaque station pour chaque période d'exploitation. Cette notion est à différencier du taux de service qui est assimilée à la stabilité de l'horaire vis à vis des perturbations de trafic.

Il est donc indispensable de prévoir des marges lors de l'élaboration des tables horaires, afin d'éviter l'apparition de problèmes comme les *effets boule de neige* lors de l'exploitation de la ligne.

— **les caractéristiques d'exploitation des trains** : les plages de vitesse commerciales, les taux d'accélération et de décélération, la durée des phases de freinage, la longueur des trains, ...

Le lecteur intéressé pourra se référer aux travaux de [13] et [14] qui ont effectué une étude détaillée des composantes permettant de définir la capacité d'une ligne, et à [15] qui fait une présentation assez exhaustive des travaux effectués sur l'évaluation de la capacité pour différentes lignes réelles ou fictives.

1.3.3 Problème de planification des trains

La résolution de l'ensemble des conflits générés par ces deux phases de planification permet de créer des tables horaires qui respectent l'ensemble des exigences d'exploitation.

Ce problème de planification ferroviaire est plus connu dans la littérature sous l'appellation *train timetabling problem* (TTP), il s'agit d'un des problèmes les plus complexes dans le domaine ferroviaire qui a fait l'objet d'un très grand nombre de publications scientifiques et de travaux de thèse.

Avant l'utilisation de l'informatique, la planification des horaires des réseaux ferroviaires était réalisée manuellement.

Les tables horaires étaient conçues par essais et erreurs et étaient alors très dépendantes de l'expérience du planificateur et de la connaissance de la ligne [16], [17],[18].

Historiquement, ce sont les méthodes mathématiques issues de la Recherche Opérationnelle (RO) qui ont été les premières à être utilisées pour résoudre le TTP [19].

1.3.3.1 Utilisation de la Recherche Opérationnelle

La Recherche Opérationnelle moderne tire son origine de la seconde guerre mondiale. À cette époque, l'armée britannique souhaitait d'une part rationaliser la logistique de l'approvisionnement en ressources de ses différentes opérations et d'autre part déterminer les emplacements optimaux d'un réseau de radars sur son territoire pour se prémunir contre d'éventuelles attaques aériennes.

L'utilisation de la RO s'est ensuite étendue et généralisée à de nombreuses applications civiles comme l'économie, la médecine, la physique,...

La Recherche Opérationnelle a également eu un rôle prépondérant dans la résolution de problèmes issus de la planification ferroviaire. L'objectif initial était simple : comment augmenter la productivité d'un réseau ferroviaire en minimisant les investissements et les aménagements à effectuer ?

La *Programmation Linéaire* (PL) est l'une des premières formalisations mathématiques de la RO.

L'éthymologie du terme Programmation Linéaire est intéressante au sens qu'initialement le mot *programme* désignait les actions de planification horaires et les choix logistiques de l'armée américaine.

C'est donc assez naturellement que la PL et plus globalement la RO ont été utilisées dans le cadre de la planification ferroviaire [15].

La programmation linéaire a ensuite été déclinée en plusieurs variantes permettant par exemple de résoudre des problèmes non linéaires ou constitués de variables entières [19], [20].

1.3.3.2 Exemples de résolution du train timetabling problem

La première tentative de résolution du TTP a été effectuée en 1971 par [21], pour concevoir des tables horaires à l'aide de méthodes d'optimisation mathématiques.

[22] a ensuite introduit en 1989 la notion de planification d'évènement périodique (*periodic event scheduling problem* (PESP)), en considérant la planification horaire de lignes de métro comme la résolution cyclique de TTP.

De nombreuses variantes du TTP ont été étudiées avec des objectifs assez différents selon les applications. [23] a ainsi proposé une méthode dérivée du PESP pour déterminer l'intervalle d'exploitation minimal d'une table horaire tandis que [24] et [25] ont proposé des modèles de tables horaires minimisant le temps d'attente des passagers en station ainsi que l'utilisation du matériel roulant.

A partir des années 1990, l'utilisation d'heuristiques [26],[27], de la logique floue [28] de méthodes de recherche [29] ou encore de techniques de recherche évolutionnaire [30], [31], [9], ont permis de proposer des approches différentes pour résoudre les problèmes de TTP et de diversifier les objectifs de leurs études.

1.4 Optimisation énergétique en milieu ferroviaire

Dans la littérature ferroviaire, l'optimisation énergétique d'une ligne de métro fait référence généralement à la poursuite de deux objectifs : la réduction de la consommation électrique globale et la réduction de la puissance maximale appelée.

Sur une ligne ferroviaire électrique, les pics de puissance électrique se produisent lorsqu'un grand nombre de trains consomment de l'énergie simultanément, autrement dit, quand plusieurs trains sont en phase de traction.

Du point de vue de l'exploitant, l'occurrence de ces pics de puissance peut générer des pénalités à payer au fournisseur d'énergie, pour cause de dépassement excessif de la puissance souscrite.

De fait la réduction des pics de puissance permettrait à l'exploitant de pouvoir abaisser la puissance souscrite de la ligne et donc le coût de l'abonnement électrique [32].

Dans le cadre de cette thèse, l'éventualité d'un arrêt de l'exploitation en cas de dépassement de la puissance maximale admissible n'est pas traitée puisque de nombreuses études dimensionnantes sont réalisées en amont de la conception du réseau électrique de traction pour éviter de telles situations.

1.4.1 Réduction des pics de puissance électrique consommée

[33] propose de modifier l'horaire de départ aux terminus des trains à l'aide de deux méthodes de résolutions : une résolution par le logiciel commercial CPLEX et une autre par une heuristique, utilisant une formulation de programmation linéaire. L'application de ces méthodes à un modèle simplifié d'une ligne du métro de Séoul permet ainsi une réduction de la puissance maximale de traction de respectivement 32% et 27%. Cependant, ces résultats sont à contraster de par la simplicité de la modélisation et la valeur très élevée du pas de temps de simulation.

[30] évoque la possibilité de réduire les pics de consommation à l'aide d'un algorithme génétique, en allongeant le temps de parcours des trains. C'est à dire en utilisant le temps de battement pour accroître la durée de certains parcours interstation. Cette méthode appliquée à une ligne du réseau S-Bahn de Berlin permet une réduction de 17% de la puissance moyenne (valeurs moyennées sur 15 minutes). Cependant, pour notre cas d'étude, cette approche n'est pas envisageable puisque le matériel roulant utilise des profils de vitesse, que l'on ne cherchera pas à modifier.

[31] utilise également un algorithme génétique pour redéfinir la durée des temps d'arrêt en station, en considérant deux cas de figures : soit des arrêts courts de 25 secondes ou des arrêts longs de 35 secondes. Des simulations menées sur une ligne du métro de Kaohsiung ont montré une amélioration de la puissance pic d'environ 30%.

1.4.2 Diminution de la consommation électrique

La littérature portant sur la réduction de consommation de lignes ferrovaires est très riche :

[9] utilise un algorithme génétique pour définir le nombre et la durée des phases d'accélération, de freinage et de marche sur l'erre de parcours interstation.

[34] utilise également un algorithme génétique pour moduler la durée des temps d'arrêt en station. Ces travaux se distinguent des autres par une modélisation très précise des paramètres électriques du réseau ferroviaire. Bien que le cas d'étude choisi reste très simpliste (ligne composée de quatre stations et quatre trains), il offre une démonstration de la validité de la démarche.

[28] met en œuvre un contrôleur flou pour modifier la durée des temps d'arrêt en station afin de maximiser le taux de réceptivité de la ligne. Ce contrôleur évalue la probabilité d'occurence des transferts d'énergie entre trains.

[35] & [36] utilisent une formulation par PL pour optimiser une table horaire de nuit pour le métro de Madrid en maximisant les périodes de synchronisation des phases de freinage et d'accélération de plusieurs trains. Afin de limiter la complexité du problème, la synchronisation est effectuée par couples de trains. Des essais sites de la méthode ont montré une diminution de 3% de la consommation électrique de la ligne.

[37] souhaite concevoir des tables horaires ayant deux objectifs : la minimisation du temps d'attente des passagers et la maximisation du temps de chevauchement entre phases d'accélération et de freinage. L'auteur prend en compte plusieurs contraintes comme l'utilisation de profils de vitesses, la régulation de trafic autour d'un intervalle

fixe ou la modification des temps d'arrêt en respectant un temps de battement minimal en terminus. La mise en œuvre de la méthode sur une ligne du métro de Pékin, a permis d'enregistrer un gain énergétique de 8% par rapport à une table horaire nominale.

Dans [38], l'auteur utilise une heuristique pour synchroniser les phases de freinage et d'accélération de deux métros consécutifs. L'une des solutions utilisée consiste à insérer des phases de marche sur l'erre. Une simulation de la méthode sur une ligne du métro de Pékin a montré une baisse de la consommation énergétique d'environ 10%.

[39] utilise un simulateur électrique pour modéliser les transferts d'énergie dans la ligne de métro. Les auteurs appliquent une modification des profils de vitesse commerciale ainsi qu'une modification des horaires de départs en station pour effectuer l'optimisation de la consommation. Cependant, l'étude étant réalisée sur une ligne composée de deux métros, son intérêt reste assez limité puisque la complexité du problème étudié est assez éloigné d'un cas réel d'exploitation.

1.4.3 Modélisation des flux de puissance

L'état de l'art qui a été effectué sur la problématique de l'optimisation énergétique a permis de mettre en lumière la nécessité de disposer d'un modèle précis des flux de puissances qui s'effectuent sur les lignes ferroviaires.

En effet, parmi les travaux cités précédemment, un grand nombre utilisent un modèle approché qui ne tient pas compte de la dynamique et de la non-linéarité du freinage récupératif, ce qui ne produit pas des résultats réalistes.

A l'inverse, [40], [31] ou encore [39] proposent des outils de simulation assez précis qui permettent d'avoir une assez bonne représentation de l'évolution des paramètres électriques de la ligne.

En outre, la connaissance des flux de puissance entre les différents éléments présents sur une ligne ferroviaire est particulièrement importante dans l'optique d'effectuer une optimisation en temps réel des paramètres de fonctionnement d'une ligne.

1.5 Conclusion

Dans ce premier chapitre,un historique des transports urbains ainsi que du contexte sociétal de l'étude a d'abord été présenté afin d'expliquer l'intérêt de faire évoluer l'exploitation des transports urbains.

La problématique de planification ferroviaire a ensuite été introduite pour définir le cadre de l'étude.

Puis un état de l'art des travaux portant sur l'optimisation énergétique en milieu ferroviaire a été dressé afin de souligner les solutions déjà explorées, mais également mettre en avant les limites de ces travaux pour réaliser une optimisation en temps réel de la consommation énergétique d'une ligne ferroviaire.

Ce chapitre a mis en évidence la nécessité d'effectuer une modélisation énergétique précise des éléments circulant sur une ligne ferroviaire afin de pouvoir calculer la consommation induite par l'exploitation et ainsi en déduire des solutions pour la minimiser.

Le chapitre suivant se focalise sur la définition d'une méthodologie permettant de mettre en œuvre un modèle énergétique d'une ligne ferroviaire puis de caractériser le

fonctionnement du freinage récupératif pour enfin en déduire une méthode de résolution permettant de calculer la quantité d'énergie qui peut réellement être renvoyée sur la ligne lors du freinage des trains en exploitation.

Chapitre 2

Modélisation d'une ligne de métro automatique

« La vie, c'est comme le métro, lorsqu'une porte s'ouvre il faut foncer. »

Fabrice Bensoussan

Sommaire

2.1 Introduction

2.1.1 Enjeux - objectifs

L'optimisation énergétique d'une ligne de métro automatique, nécessite d'abord de comprendre comment les rames de métro sont alimentées en énergie et comment les échanges de puissances s'effectuent entre les trains et les sous-stations d'alimentation. L'objectif de ce chapitre est d'établir un modèle électrique d'une ligne de métro pour nous permettre d'estimer la consommation instantanée des véhicules en circulation, ainsi que les échanges d'énergie qui se produisent entre véhicules lors des phases d'arrivée en station. Ce modèle devra en outre être assez simpliste pour permettre une mise en œuvre en temps-réel.

Dans ce chapitre, les principaux éléments qui constituent une ligne de métro automatique sont passés en revue : le matériel roulant, les barres de guidage qui permettent d'alimenter les rames en énergie et les sous-stations d'alimentation.

Puis, un modèle cinétique et électrique du matériel roulant est présenté afin d'établir le lien entre le déplacement des véhicules et les flux de puissance électrique que cela engendre. Le fonctionnement du freinage électrique est également caractérisé, pour permettre de comprendre les conditions de transformation de l'énergie cinétique du véhicule en énergie électrique.

Ensuite, une modélisation des sous-stations d'alimentation et des barres de guidage servant au transport de l'énergie sur la ligne est menée, afin d'obtenir un modèle suffisamment précis pour envisager de déterminer les flux de puissance qui s'opèrent au sein du réseau lors de l'exploitation de la ligne de métro.

Enfin, des méthodes de résolution sont mises en œuvre pour déterminer, à chaque pas de temps de simulation, les caractéristiques électriques de chacun des éléments constituant la ligne et d'en déduire les échanges d'énergie qui s'opèrent au sein du réseau électrique.

2.1.2 Terminologie

Avant de débuter la description des sous-systèmes qui composent une ligne de métro automatique, il convient de définir la terminologie propre au matériel roulant.

- Une voiture consiste en une caisse supportant des équipements et reposant sur deux bogies.
- Un doublet est constitué de deux voitures reliées par une barre semi-permanente.
- Un triplet est constitué de trois voitures reliées par une barre semi-permanente.
- Un véhicule est la plus petite unité utilisable en exploitation ; il peut s'agir d'un doublet ou d'un triplet.
- Un train peut être constitué d'un ou de deux doublets.

La figure 2.1 présente le schéma d'une voiture de type NEOVAL ainsi que ses principales dimensions.

2.1.3 Spécificités de l'étude

Dans cette étude, les systèmes de métro automatique développés par l'entreprise Siemens (anciennement MATRA), à savoir les systèmes VAL et Néoval sont particulièrement étudiés.

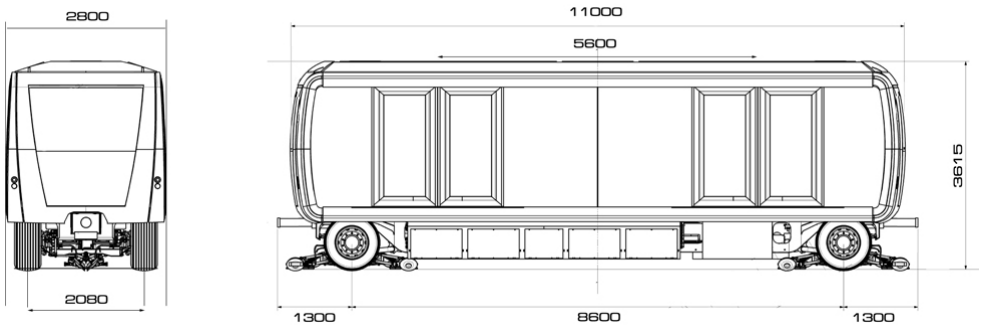


FIGURE 2.1 – Schéma d'une voiture de type NEOVAL.

Le VAL, acronyme pour Villeneuve d'Ascq-Lille, puis par la suite Véhicule Automatique Léger, a été mis en service pour la première fois en 1983 pour relier les villes de Lille et Villeneuve d'Ascq, sur ce qui constituait la première ligne de métro entièrement automatisée au monde. Le concept du VAL avait initialement trois grands objectifs :

— Offir une haute qualité de service, avec une vitesse commerciale et une fréquence d'exploitation élevées.

— Pouvoir s'insérer dans un contexte urbain qui impose notamment des contraintes sur la déclivité et la courbure des voies.

— Permettre une conduite automatisée pour réduire les coûts d'exploitation.

En 1968, le cahier des charges du VAL imposait ainsi de pouvoir transporter 6000 passagers par heure et par direction en heure creuse.

Actuellement, douze lignes de VAL sont en service à travers le monde.

En 2006, Siemens et Lohr Industrie s'associent pour créer le Néoval. Ce dernier a pour but de compléter l'offre VAL, en proposant des véhicules plus larges (2,65m à 2,80m), une modularité plus complète (1 à 6 voitures), une capacité de transport accrue jusque 30000 passagers par heure et par direction, la prise en compte des normes internationales et une plus grande compétitivité économique pour l'exploitant. En outre, certaines versions de Néoval auront également la capacité de pouvoir fonctionner sans alimentation extérieure sur des portions de ligne grâce à l'ajout de systèmes de stockage d'énergie embarqués.

Le Néoval intègre également un système de communication CBTC. Le CBTC (*Communication Based Train Control*) est une norme IEEE[1] qui s'applique à l'automatisation du contrôle-commande de la marche des trains et spécifie les exigences fonctionnelles, les exigences de performance, les critères d'intervalle (la fréquence de passage), les critères de sécurité système et les critères de disponibilité système.

Un système CBTC est un ATC (*Automatic Train Controller*) qui permet de gérer le fonctionnement de la ligne en localisant avec une forte résolution les trains de façon indépendante des circuits de voie. Ce système se caractérise également par l'utilisation d'une transmission de données, entre le sol et les trains, continue, bi-directionnelle et à haut débit et l'installation d'ordinateurs à bord des trains et au sol qui effectuent des traitements de sécurité.

1. *Institute of Electrical and Electronics Engineers*

2.2 Présentation du réseau de traction

Dans cette section, les différents éléments qui composent le réseau de traction d'une ligne de métro sont détaillés. Seuls les équipements utiles à la modélisation et à la compréhension du système sont explicités.

La figure 2.2 présente le schéma d'alimentation électrique d'une ligne de métro, ainsi que les niveaux de tension vus par les différents équipements. Les Postes de Livraisons (PLv) distribuent l'énergie électrique aux Postes de Redressement (PR) qui alimentent la ligne de métro et aux Postes Eclairage et Force (PEF) qui alimentent les stations.

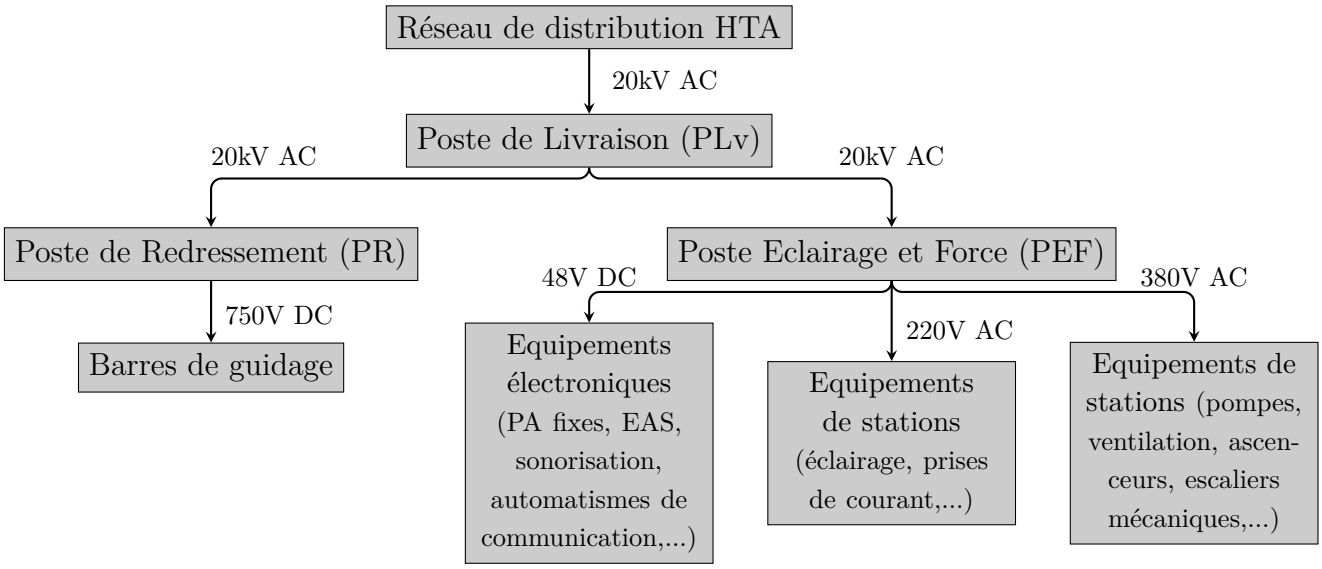


FIGURE 2.2 – Schéma de l'alimentation générale d'une ligne de métro.

Les termes *sous-station*, *sous-station d'alimentation*, *sous-station de traction* et *poste de redressement* font référence au même équipement.

Les trois premiers termes sont les dénominations classiquement utilisées dans la littérature ferroviaire et s'appliquent aussi bien à des réseaux à courants alternatifs que des réseaux à courants continus, tandis que le dernier est un terme spécifique aux réseaux électriques ferroviaires de type continu et est communément utilisé au sein de Siemens.

Pour plus de clarté, les termes *sous-station d'alimentation*, ou *sous-station* sont employés dans la suite du développement.

De plus, l'emploi du terme *rail d'alimentation* sera privilégié au détriment de celui de *barre de guidage*, afin de ne pas entrainer de confusion avec le rail de guidage présent sur les systèmes Néoval.

2.2.1 Sous-station d'alimentation

2.2.1.1 Caractéristiques des sous-stations d'alimentation

Les sous-stations d'alimentation ont deux fonctions principales : l'alimentation des circuits de chauffage de voie [2] et la distribution d'énergie aux sections de ligne auxquelles elles sont connectées.

Dans chaque sous-station la tension issue du réseau HTA (généralement, de l'ordre de 20 kV à 50Hz) est transformée puis redressée en 750 V continu, pour ensuite être transmise aux rails d'alimentation qui vont électrifier l'ensemble de la ligne.

En fonctionnement nominal, une sous-station peut délivrer un courant de 2400A sous 750V, tandis que sa tension à vide avoisine les 825V.

Les sous-stations alimentées en triphasé, sont essentiellement constituées d'un transformateur triphasé 20 kV / 750 V d'une puissance nominale secondaire de 2000kVA, d'un redresseur dodécaphasé d'une puissance nominale de 1800kW, de disjoncteurs et de sectionneurs à courant continu.

L'intérêt majeur d'un redresseur dodécaphasé est d'atténuer les harmoniques de rang peu élevé mais d'amplitudes importantes pour améliorer la qualité du signal électrique fourni.

2.2.1.2 Problématique d'implantation

La répartition des sous-stations d'alimentation le long de la ligne est définie en fonction des exigences de l'exploitant, de manière à garantir un certain taux de service en mode dégradé, c'est à dire, lors de la perte d'une sous-station.

En mode dégradé, l'implantation des sous-stations doit permettre la circulation d'un nombre assez important de rames à un intervalle donné, tout en n'excédant pas les limites de puissances délivrées par les sous-stations.

Cependant, dans un contexte urbain, les choix d'implantations des sous-stations sont largement dictés par la localisation des stations.

De ce fait, les distances entre les sous-stations peuvent être assez élevées, ce qui a pour effet de provoquer d'importantes chutes de tension. Pour cette raison, des feeders [3] sont installés en parallèle des voies pour limiter ces chutes de tension.

Sur le plan de la ligne de Turin (figure 2.3), les stations comportant des sous-stations sont encadrés.

2.2.2 Feeders

Les feeders sont des conducteurs placés le long des rails d'alimentation pour limiter les chutes de tension dans ces derniers. L'ajout de feeders permet de réduire la résistivité des rails.

Dans l'équation (2.1), la chute de tension ΔU_{AB} entre les points A et B est proportionnelle à la résistivité ρ, à la longueur d_{AB} et à la section S du câble d'alimentation

2. Dans ce manuscrit, le système de chauffage de voie n'est pas développé car il ne présente pas un intérêt majeur pour le fonctionnement de la ligne de métro dans des conditions normales d'exploitations. Leur principale fonction étant d'assister au dégivrage des voies aériennes.

3. voir section 2.2.2

ainsi que du courant I_{AB} circulant entre A et B.

$$\Delta U_{AB} = \rho . \frac{d_{AB}}{S} . I_{AB} \tag{2.1}$$

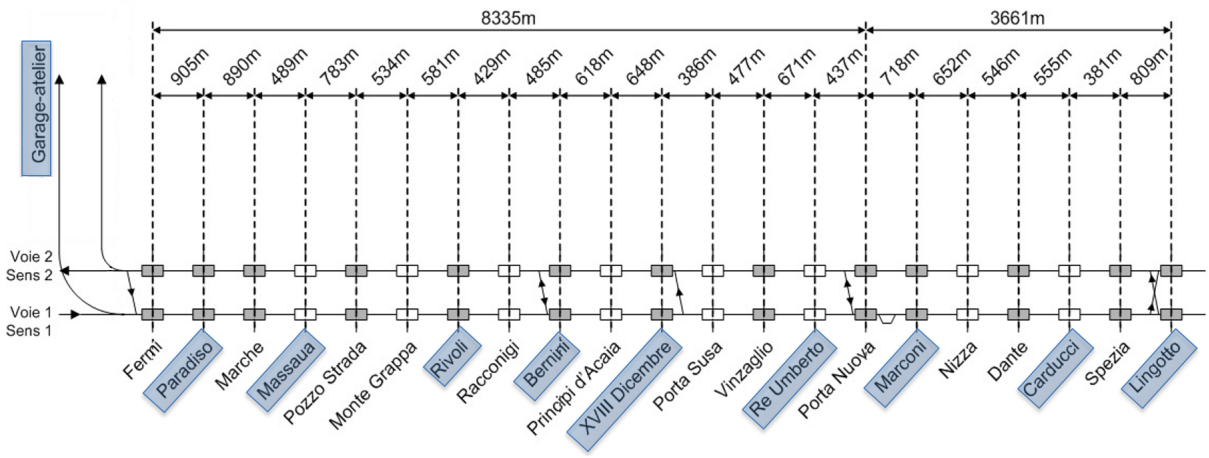


FIGURE 2.3 – Plan des stations et sous-stations d'alimentation de la ligne de métro de Turin.

2.2.3 Rails d'alimentation et système de guidage

Contrairement à certaines lignes ferroviaires ou la captation du courant se fait par contact avec une ligne aérienne, les métros de type VAL et Néoval sont électrifiées par l'installation de conducteurs placés parallèlement à la piste de roulement.

La captation du courant se fait donc via deux rails d'alimentation positif et négatif. Le rail positif est alimenté en 750V, tandis que le rail négatif assure le retour du courant.

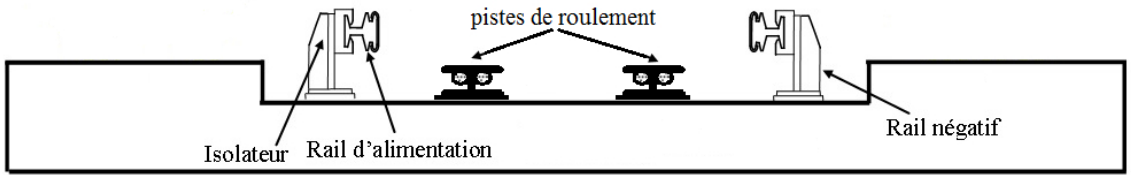


FIGURE 2.4 – Vue en coupe de la voie d'une ligne VAL.

Dans un système VAL (figure2.4), chaque voiture est constituée d'une caisse reposant par l'intermédiaire d'une suspension sur deux essieux équipés de pneumatiques guidés par quatre roues de guidage.

Les roues de guidage placées aux quatre coins de l'essieu permettent d'orienter la rame dans les courbes et de la maintenir dans l'axe de la voie (figure 2.4).

Dans le système Néoval (figure 2.5), un rail de guidage central est installé pour guider les trains et servir de masse du réseau électrique.

La combinaison du guidage par rail central et de l'utilisation d'essieux orientables permet de réduire les efforts sur le système de guidage en redirigeant les efforts latéraux et verticaux sur les pneumatiques.

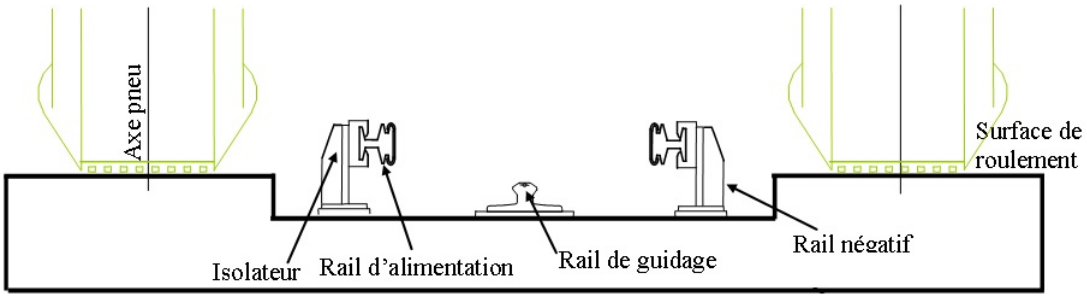


FIGURE 2.5 – Vue en coupe de la voie d'une ligne Néoval.

Ainsi, le rail de guidage est peu sollicité, puisque la quasi intégralité du poids des véhicules est assumé par les pneus.

Enfin, il est à noter que la mise à la terre de l'ensemble du réseau de traction est réalisée en un seul point, appelé puits de terre et que toutes les liaisons de terre sont interconnectées afin d'assurer l'équipotentialité du réseau électrique.

2.2.4 Coffrets de Surveillance du Potentiel Négatif

Dans un système VAL, la masse des trains est reliée au rail négatif. Le réseau de traction est donc dimensionné de manière à ce que le potentiel du rail négatif soit le plus souvent compris entre les valeurs de + 60V et - 60V.

Cependant, dans certains modes de fonctionnement, il peut arriver que le potentiel de ce rail excède cette limite.

C'est pourquoi, afin d'assurer la sécurité des voyageurs montant ou descendant des trains, des Coffrets de Surveillance du Potentiel Négatif (CSPN) sont installés dans chaque station.

Ces dispositifs assurent un contrôle permanent de la valeur du potentiel du rail négatif et procèdent à la mise à la terre automatique du rail négatif lorsque sa tension excède une valeur limite : ± 60V (dans le cas où une rame est en station) ou ± 200V (dans le cas où aucune rame n'est en station).

2.3 Présentation du matériel roulant

2.3.1 Alimentation électrique du matériel roulant

La figure 2.6 représente le schéma électrique simplifié du système de propulsion et freinage d'un véhicule (PBS pour *Propulsion and Braking System*), sur lequel sont indiqués les principaux éléments permettant l'alimentation en courant des moteurs synchrones.

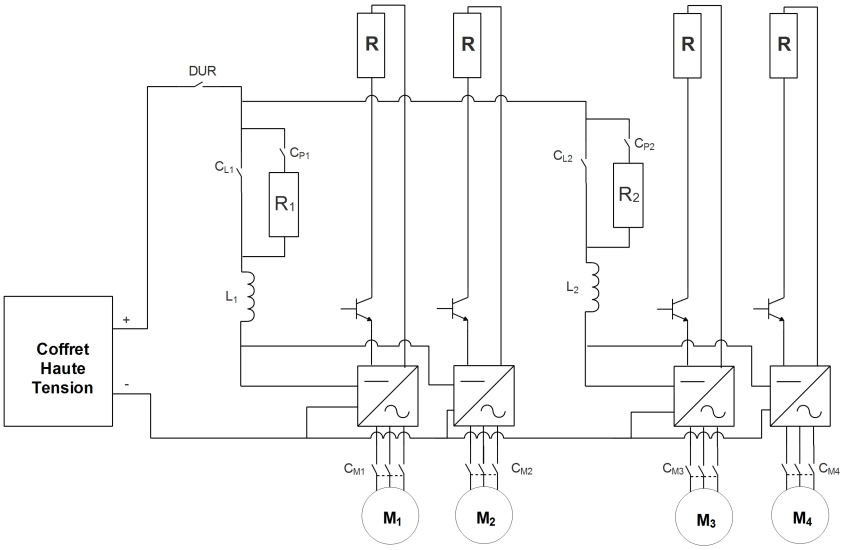


FIGURE 2.6 – Schéma d'alimentation des moteurs d'un véhicule.

Une fois délivrée par les sous-stations, la tension continue de la ligne est d'abord filtrée par une inductance et un condensateur connectés en entrée d'onduleurs triphasés à transistors de type IGBT [4] qui alimentent les phases des moteurs en courant.

La propulsion de chaque voiture de type VAL 208 et Néoval est assurée par quatre moteurs synchrones triphasés à aimants permanents, chacun alimenté par son propre onduleur. Chaque roue d'une voiture est ainsi équipée d'un moteur synchrone dédié.

Les performances des moteurs imposent le respect d'une tension d'alimentation comprise entre 450V et 960V. Cependant, la valeur basse retenue comme critère de dimensionnement est 550V.

Les IGBT des onduleurs sont pilotés par l'électronique de traction en fonction de la référence de courant qui détermine également le couple moteur. La séquence de pilotage des IGBT est quant à elle déterminée par la position relative du rotor par rapport au stator, du sens de marche et de l'ordre de traction ou freinage.

Il est à noter que suivant les lignes de métro étudiées, la dissipation du freinage électrique est réalisée tantôt dans un rhéostat embarqué, tantôt dans un banc de charge fixe placé en un point de la ligne. La première solution est généralement mise en oeuvre dans les systèmes Néoval tandis que la seconde est privilégiée dans les systèmes VAL.

4. Un IGBT, de l'anglais, *Insulated Gate Bipolar Transistor*, est un semi-conducteur de la famille des transistors. Sa popularité vient de ses faibles pertes de conduction et d'un faible coût énergétique de commande.

Ainsi, la figure 2.6 représente plutôt le fonctionnement d'un système Néoval, où chaque onduleur possède son propre rhéostat de freinage auto-ventilé en toiture (noté R).

2.3.2 Description de la chaine de traction

La chaine de traction d'un train VAL / Néoval est composée des éléments présents sur la figure 2.7.

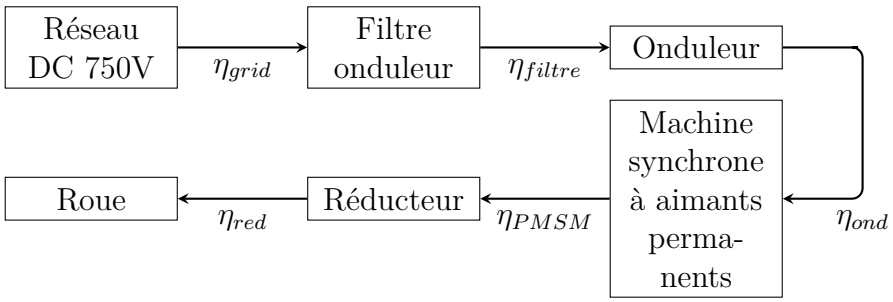


FIGURE 2.7 – Composition de la chaine de traction d'un métro de type VAL.

En première approximation, la chaine de traction du VAL 206 est considérée équivalente à celle du VAL 208.

Les coefficients de rendement de chaque élément de la chaine de traction sont rappelés dans le tableau 2.1.

	Val 208	**Néoval**
Rendement des rails d'alimentation η_{grid}	0,99	
Rendement du filtre de l'onduleur η_{filtre}	0,99	
Rendement de l'onduleur η_{ond}	0,98	
Rendement moyen de la machine synchrone η_{PMSM}	0,95	
Rendement du réducteur η_{red}	0,965	0,95

TABLEAU 2.1 – Caractéristiques des machines tournantes.

2.3.3 Principe de fonctionnement du système de traction d'une voiture

le schéma de principe 2.8 illustre la chaine de contrôle du système de traction d'un véhicule.

Lors de l'exploitation en mode nominal, la vitesse du véhicule est régulée par un asservissement bouclé réalisé par le Pilote Automatique (PA) ou Automatic Train Control (ATC).

Ce dernier reçoit de la voie un signal correspondant à la vitesse de palier à atteindre. Cette référence est ensuite comparée à la vitesse réelle du véhicule mesurée par une génératrice tachymétrique.

L'unité de contrôle du véhicule (VCU) reçoit la consigne d'effort de l'ATC et donne l'ordre de traction aux différentes unités de contrôle de la traction (TCU). Une TCU est dédié à chaque essieu.

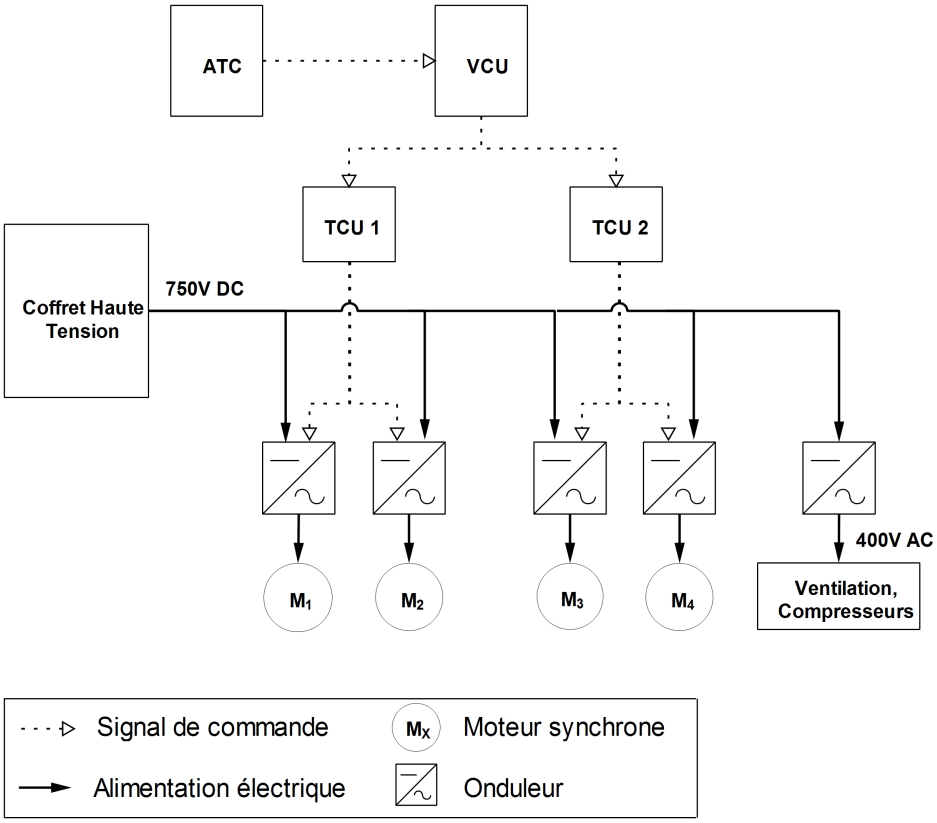


FIGURE 2.8 – Principe de fonctionnement du système de traction d'un véhicule.

2.3.4 Conditions d'utilisation du système de freinage

Le tableau 2.2 récapitule les cas d'utilisation des freins mécanique et électrique en fonction des conditions d'exploitation.

— Le frein de service est utilisé pour permettre le ralentissement du véhicule en fonction du profil de vitesse ; il s'effectue par la conjugaison du frein mécanique et du frein électrique pour fournir un freinage contrôlé. Dans ce mode, priorité est donnée au frein électrique.

— Le frein d'urgence permet au véhicule de s'arrêter en cas d'urgence ; pour des raisons de sécurité, il est entièrement fourni par le frein à friction et est prioritaire sur les autres actions.

— Le frein de stationnement permet d'assurer la sécurité du stationnement de secours d'un véhicule ayant reçu la commande d'arrêt. Il est exclusivement effectué par le frein mécanique.

2.3.5 Plage d'application du freinage conjugué

Lorsqu'un train est en cours de freinage, celui-ci va voir sa tension d'alimentation augmenter due à la réinjection de courant sur le réseau électrique. Si le courant renvoyé par le train est consommé par d'autres trains, sa tension d'alimentation reste à un niveau acceptable mais si une partie de ce courant ne peut être consommé, la tension d'alimentation du train augmente jusqu'à atteindre une valeur limite qui entraine une conjugaison du freinage électrique avec du freinage mécanique.

	Frein mécanique à friction	**Frein électrique**
Freinage de service	✓ (uniquement en mode dégradé)	✓
Freinage d'urgence	✓	✗
Arrêt en station et freinage en côte	✓	✗
Frein de stationnement	✓	✗

TABLEAU 2.2 – Récapitulatif des cas d'utilisation des freins mécanique et électrique.

Lorsque la tension de la ligne dépasse la valeur seuil, le freinage électrique et le freinage rhéostatique/mécanique sont combinés pour maximiser la récupération du freinage électrique tout en respectant le confort des voyageurs.

La figure 2.9 présente l'évolution de la consigne de courant en fonction de la tension d'alimentation. La consigne maximale de courant est applicable entre 450V et 860V, tandis qu'entre 860V et 960V, la consigne maximale admise est réduite linéairement à zéro. Dans la suite du manuscrit, I_{max} sera pris comme le courant maximal généré/absorbé par les moteurs d'un train et la notation suivante est adoptée : $E_- = 860\text{V}$, $E_+ = 960\text{V}$.

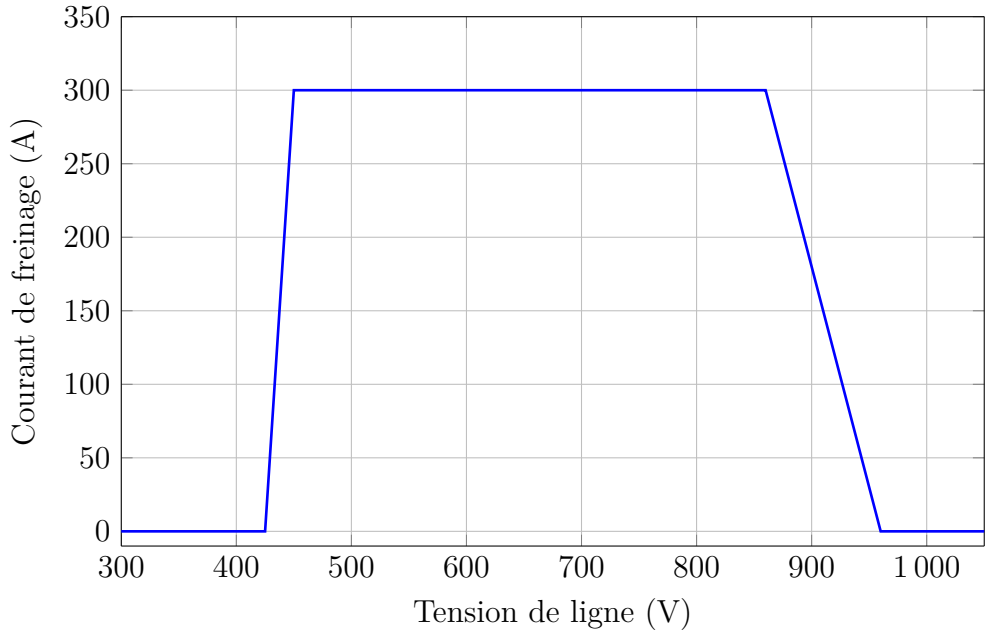


FIGURE 2.9 – Evolution de la consigne du courant moteur en phase de freinage.

2.3.5.1 Cas particulier du système Néoval

A la différence du VAL, où le frein électrique est inhibé en dessous d'une certaine valeur de vitesse, le frein de service du Néoval est assuré par des équipements électriques via les roues motrices, avec récupération de l'énergie, jusqu'à l'arrêt complet du

véhicule. De plus, le véhicule est équipé de résistances embarquées afin de maintenir le freinage électrique opérationnel même dans le cas où la ligne ne serait pas réceptive, c'est à dire lorsqu'il n'y a pas assez de trains capables d'absorber l'énergie générée par les trains en phase de freinage. Ces rhéostats de freinage sont calibrés pour supporter la complète décélération du véhicule, de la vitesse maximale à l'immobilisation.

En outre, dans le système Néoval, le frein mécanique n'est utilisé pour le freinage de service que pour des cas bien particuliers comme la perte de l'alimentation HTA ou une panne partielle du frein électrique.

L'utilisation quasi-systématique du frein électrique dans le système Néoval présente plusieurs avantages :

— une réduction de l'usure des garnitures de frein par rapport à un système VAL classique,

— une baisse de la pollution aux particules fines émises par la friction métallique des freins mécaniques,

— l'adoption d'une démarche éco-responsable grâce à la récupération du freinage électrique,

— une conduite des trains plus fluide avec un confort des passagers accru et une meilleure précision dans le contrôle des différentes phases de marche des trains.

2.3.6 Equipements auxiliaires

Le circuit auxiliaire des voitures est composé de deux sous-réseaux électriques. Un premier sous-réseau triphasé à 400V alimente la ventilation ainsi que les compresseurs hydrauliques et pneumatiques. Le second sous-réseau alimente en 24V continu le chargeur de batterie, l'éclairage intérieur, les portes palières, la signalisation extérieure,...

Une batterie est également présente dans chaque voiture en cas de perte de la haute tension ou de panne des convertisseurs. Dans cette éventualité, la batterie alimente les équipements de secours comme la ventilation, l'éclairage, la sonorisation et système de communication, le contrôle des freins mécaniques, les portes palières, quelques équipements du pilote automatique, et d'autres dispositifs nécessaires aux opérations de secours. Ce mode de fonctionnement peut être maintenu pendant une période supérieure à 60 minutes.

2.4 Modélisation énergétique du matériel roulant

La modélisation de lignes ferroviaires a fait l'objet de nombreux travaux dans la littérature scientifique.

Dès 1987, [41] propose de simuler le déplacement de plusieurs trains sur une ligne ferroviaire, puis met en évidence la variation de la tension de ligne en fonction de la position et de la consommation électrique des différents trains.

Cette étude est l'une des premières à proposer une modélisation détaillée des différents éléments présents sur la ligne pour en estimer la consommation énergétique.

Depuis, la plupart des modélisations de lignes ferroviaires adoptent une méthodologie analogue axée principalement autour de l'étude de trois sous-systèmes : le dé-

placement des trains sur la ligne, les efforts de traction qui en résultent et le système d'alimentation électrique de la voie [42].

2.4.1 Modélisation du déplacement des trains

Selon le niveau de précision souhaité, il existe classiquement deux approches pour modéliser le déplacement des trains sur une ligne ferroviaire : l'approche temporelle et l'approche épisodique (le terme *épisodique* provient de la traduction du terme anglais *event-based*) [42].

2.4.1.1 Approche épisodique

L'approche dite épisodique, définit le mouvement des trains par une séquence d'événements caractéristiques tels que l'arrivée et le départ de station.

L'intérêt majeur de cette approche est de pouvoir avancer l'horloge de simulation jusqu'à l'occurence de l'événement suivant. Cette méthode a notamment été utilisée dans [24], [43] et [44], mais le niveau de précision n'est pas suffisant pour effectuer une gestion énergétique d'une ligne de métro.

Cette approche a plutôt pour objectif d'effectuer une étude macroscopique des tables horaires et de définir des algorithmes de régulation de trafic.

2.4.1.2 Approche temporelle

L'approche temporelle consiste à discrétiser l'horizon de simulation afin d'étudier le mouvement des trains à chaque pas de temps.

Mathématiquement, il est possible de représenter cette approche par le système d'équations (2.2), où $X(t), V(t), \gamma(t)$, représentent respectivement la position, la vitesse et l'accélération du train à l'instant t, tandis que δt représente le pas de discrétisation choisi.

De cette manière à chaque pas de temps la vitesse et la position des trains sont mises à jour pour simuler leur déplacement sur la ligne. Cette méthode est celle classiquement utilisée dans les travaux d'estimation de consommation énergétique.

$$\begin{cases} V(t+\delta t) = V(t) + \gamma(t).\delta t \\ X(t+\delta t) = X(t) + V(t).\delta t \end{cases} \tag{2.2}$$

2.4.2 Modélisation mécanique du matériel roulant

La modélisation des efforts de traction se fait quasi-systématiquement par application de la deuxième loi du mouvement de Newton, appelée aussi Principe Fondamental de la Dynamique (PFD) [45],[46], [47], [48], [49], [50].

Cette loi permet d'exprimer les efforts à la jante, et par extension la puissance électrique consommée par les trains en fonction des paramètres d'exploitation tels que la vitesse commerciale, l'accélération, le profil de la ligne,...

Cette méthode de modélisation est populaire pour sa simplicité d'implémentation et le fait que les paramètres d'exploitation nécessaires au calcul des efforts sont généralement connus, ou facilement déterminables.

Dans le cas des lignes de métro automatisées par Siemens, pour chaque interstation, des diagrammes de marche type d'une rame sont définis à vitesse nominale et à vitesse maximale.

Ces deux types de marche permettent d'avoir une plus grande flexibilité d'exploitation en cas d'aléas, ou en période de pointe.

Les figures 2.10, 2.11 et 2.12 présentent un exemple de marche type sur un parcours interstation avec un profil de vitesse, un profil de position et un profil d'accélération.

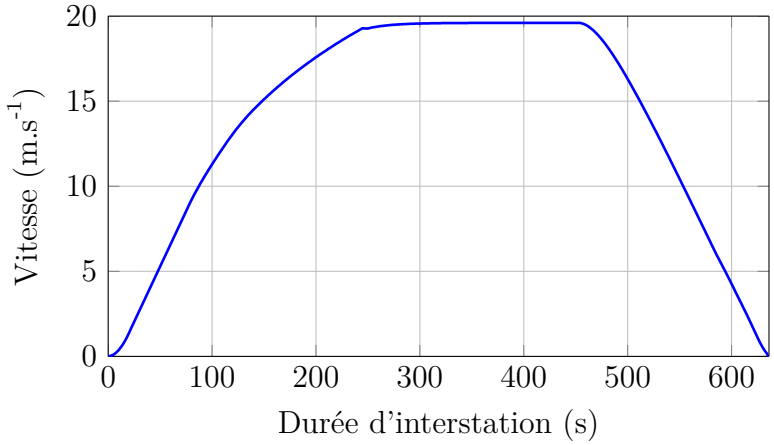


FIGURE 2.10 – Profil de vitesse pour un parcours interstation (vitesse nominale).

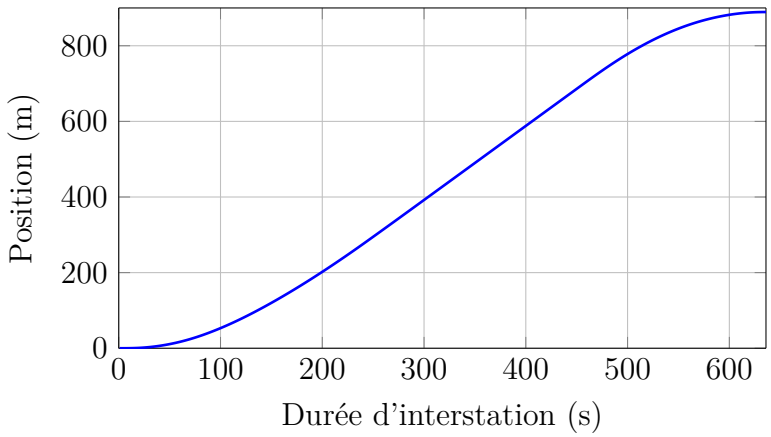


FIGURE 2.11 – Profil de position pour un parcours interstation (vitesse nominale).

Une marche type définit la vitesse de déplacement [5] d'un train sur un parcours interstation sous l'hypothèse d'une exploitation sans aléa. Ainsi, pour chaque parcours interstation, il y a une correspondance entre la position d'un train et sa vitesse de déplacement. Par extension, il est également possible de déterminer la durée de parcours d'une interstation à vitesse nominale et maximale.

Pour établir le modèle mécanique du matériel roulant, il est nécessaire d'identifier les différentes forces qui s'opposent à l'avancement d'un train. Ces efforts sont principalement de trois types : les efforts liés aux frottements (contact roues-rails et frottements aérodynamiques), les efforts liés à la pente de la voie et les efforts liés à l'accélération du train (2.3).

5. La vitesse commerciale est limitée soit par les contraintes géométriques de la ligne soit par la vitesse maximale admissible sur la portion de voie considérée.

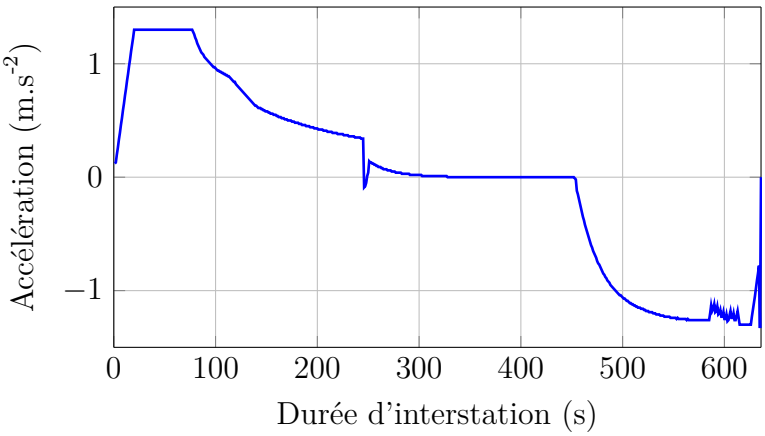


FIGURE 2.12 – Profil d'accélération pour un parcours interstation (vitesse nominale).

$$F_{avancement} = F_{res} + F_{pente} + F_{acc} \tag{2.3}$$

Les forces de frottements sont déterminées par l'équation empirique (2.4), où A, B et C représentent respectivement le coefficient de frottements secs (proportionnel à la charge par essieu, en N), le coefficient de frottements visqueux (en $N.s.m^{-1}$) et le coefficient de résistance aérodynamique (en $N.s^2.m^{-2}$), tandis que V et W représentent la vitesse du véhicule et la vitesse du vent (en $m.s^{-1}$).

$$F_{res}(t) = A + B \cdot V(t) + C \cdot (V(t) - W(t))^2(t) \tag{2.4}$$

L'expression algébrique des coefficients précédents est rappelée dans le tableau 2.3 [6], où n représente le nombre de voitures par train.

	Val 208	**Néoval**
A	1400*n +0,1*M	700*n +0,1*M
B	75*n	38*n
C (accélération / souterrain)	-16,5	
C (décélération / souterrain)	-9	

TABLEAU 2.3 – Valeurs des coefficients aérodynamiques.

Pour plus de simplicité, nous faisons ici l'hypothèse que les lignes étudiées sont souterraines ($W = 0$), que les courbes de la voie sont suffisamment faibles pour négliger les efforts dues aux courbes et que la force d'arrachement est également négligeable par rapport aux autres efforts.

Les efforts dus à la déclivité sont exprimés par la relation (2.5), où α, M et g correspondent respectivement au profil topographique de la ligne, à la masse du train passagers inclus et à l'accélération de la pesanteur

$$F_{pente}(t) = M.g.\alpha(t) \tag{2.5}$$

6. Le VAL 208 étant une évolution du VAL 206, nous présenterons ce premier modèle ainsi que le Néoval ; et seuls les trains composés d'un doublet de voitures sont considérés.

Les efforts dus à l'accélération sont quant à eux exprimés par la relation (2.6). Le coefficient $\eta_{inertie}$ permet de prendre en compte l'inertie des masses tournantes tandis que γ désigne l'accélération du train considéré.

$$F_{acc}(t) = \eta_{inertie}.M.\gamma(t) \tag{2.6}$$

2.4.3 Modélisation électrique du matériel roulant

Les efforts de résistance à l'avancement définis précédemment s'appliquent à la roue. Pour en déduire la puissance électrique consommée ou générée par un train, il nous est nécessaire de considérer le rendement global de la chaine de traction η évoqué dans le tableau 2.1 afin d'appliquer le PFD (2.7). Le terme a vaut 1 en phase de freinage et -1 en phase d'accélération tandis que P_{aux} permet de prendre en compte la puissance électrique consommée par les auxiliaires

$$P_{elec}(t) = \eta^a \cdot V(t) \cdot (F_{acc}(t) + F_{pente}(t) + F_{res}(t)) + P_{aux} \tag{2.7}$$

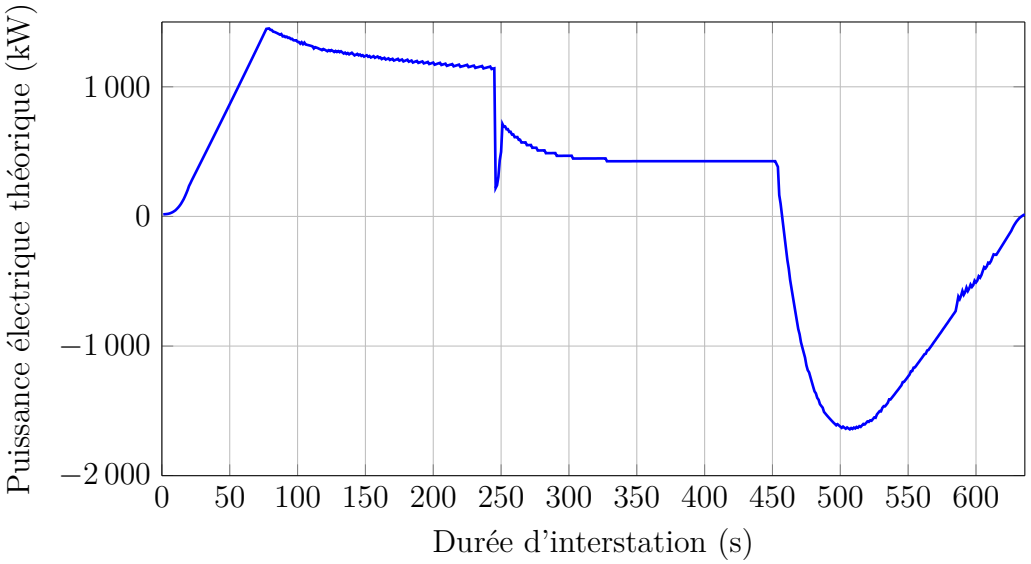


FIGURE 2.13 – Profil de puissance électrique consommée/générée par un train sur un parcours type.

Il est alors possible de définir pour chaque interstation, un profil type de puissance électrique. Un exemple de profil type de puissance, issu de la marche type présentée dans les figures 2.10, 2.11 et 2.12, est donné en figure 2.13.

Le déplacement d'un train sur une interstation est caractérisé par plusieurs phases distinctes :

— une phase d'accélération, caractérisée par une forte hausse de la puissance consommée. Cette phase d'accélération est elle même composée de deux sous-phases :

- une phase d'accélération dite à *couple constant*[7], typiquement entre 0 et 26 $km.h^{-1}$, afin de respecter les contraintes techniques qui portent sur le courant maximal admissible par les moteurs et la limite d'adhérence entre les roues et la piste de voie ;

7. Ces phases d'accélération sont particulièrement visibles sur les enregistrements d'un véhicule en exploitation présentés en figure 2.14

  - une phase d'accélération dite à *puissance constante*[7], entre 26 et 80 $km.h^{-1}$ (vitesse maximale des matériels roulants VAL et Néoval) ;

— une phase de maintien en vitesse est caractérisée par une chute de la consommation et se termine lorsque le train débute sa phase d'accostage ;

— une phase de freinage à couple constant se produit lorsque le train arrive en station. Durant cette phase, le moteur du train devient générateur, et ce couple est contraint par les caractéristiques des freins et l'adhérence des roues ;

— une phase d'arrêt en station ;

Des phases de marche sur l'erre peuvent également se produire sur un parcours interstation selon la déclivité de la voie et les contraintes opérationnelles portant sur le temps de parcours.

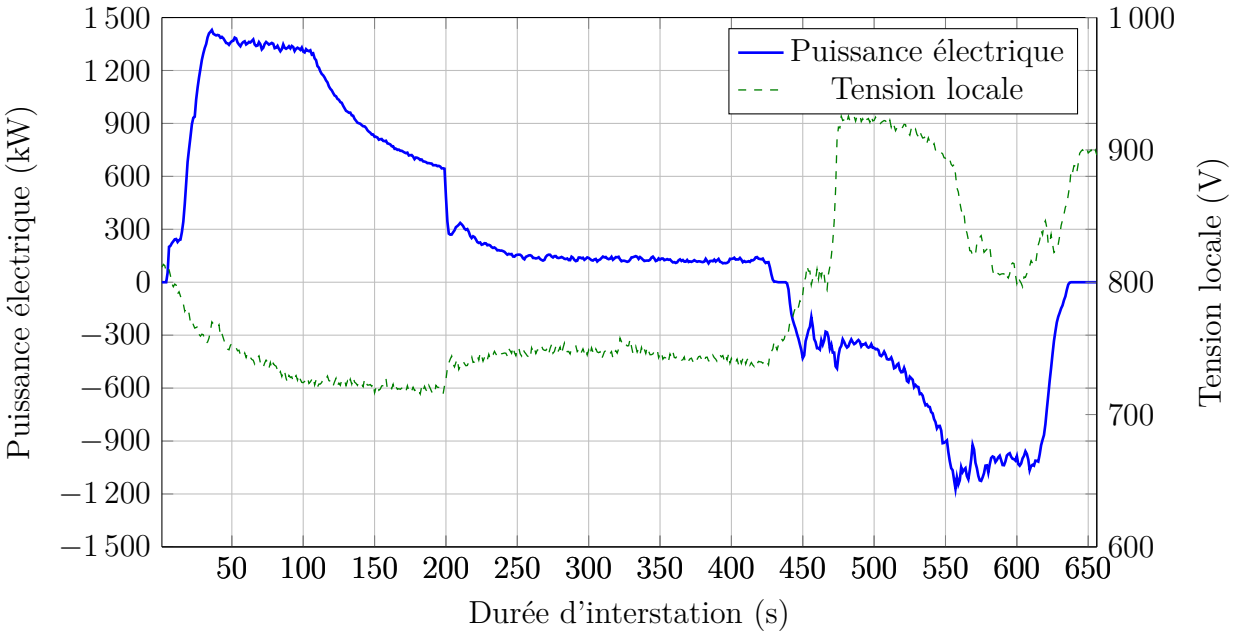


FIGURE 2.14 – Evolution de la tension locale en fonction de la puissance électrique consommée/renvoyée par un train.

Dans le cadre d'essais effectués sur la ligne de Turin, les caractéristiques électriques instantanées d'un véhicule en exploitation ont pu être enregistrés.

La figure 2.14 montre l'évolution de la tension locale en fonction de la puissance électrique consommée/renvoyée par le train durant son parcours interstation. Le parcours interstation est le même que celui évoqué précédemment.

Lors de la phase de traction, la valeur de la tension perçue par le train reste proche de la tension nominale d'exploitation, cependant, lors de la phase de freinage, la valeur de la tension locale augmente nettement. L'allure de la courbe de la puissance électrique est alors très éloignée de l'allure théorique (figure 2.13).

En effet, lors de la phase de freinage, les trains à proximité n'étaient pas en mesure de consommer la totalité de l'énergie renvoyée, ce qui a eu pour effet de faire augmenter le potentiel de la ligne et ainsi de dissiper une partie de la puissance électrique de freinage.

2.4.3.1 Conventions de modélisation

L'étude des figures 2.12, 2.13 et 2.14 montre une forte similitude de forme entre le profil d'accélération et le profil de puissance électrique qui en découle durant la phase de traction. En traction, un train est donc assimilé à une source de courant idéale absorbant de l'énergie sur le réseau (figure 2.15).

En revanche, en phase de freinage, le modèle électrique du train doit être modifié pour intégrer la notion de dissipation de l'énergie du freinage lorsque le renvoi d'énergie sur la ligne n'est pas possible. Pour ce faire, un train est assimilé soit à une source de tension en série avec un rhéostat de freinage, soit à une source de courant en parallèle avec un rhéostat de freinage. Les deux modèles sont électriquement valables, mais la première formulation est plus propice à l'analyse nodale qui sera effectuée dans la suite des travaux (figure 2.16).

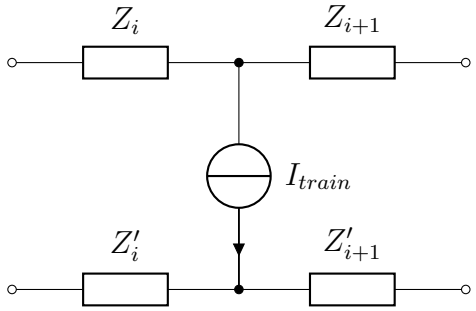
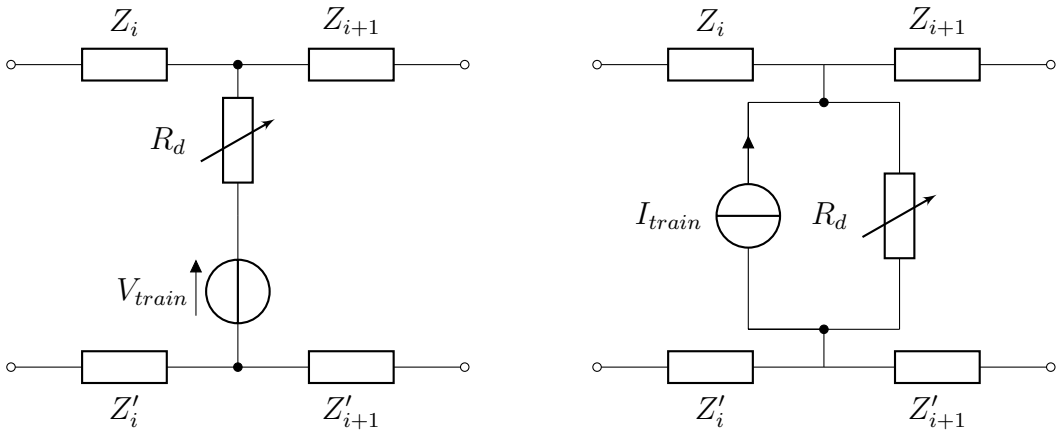


FIGURE 2.15 – Modèle électrique d'un train en traction ou en freinage purement électrique.



(a) Equivalent de Thévenin du train

(b) Equivalent de Norton du train

FIGURE 2.16 – Modèle électrique d'un train en freinage conjugué

2.4.3.2 Modélisation du rhéostat de freinage

En considérant la caractéristique de décroissance du courant de freinage présentée à la figure 2.9, l'expression de la résistance de dissipation R_d est déterminée par interpolation linéaire grâce au développement explicité dans l'équation (2.8) ; où P_{train} est assimilée à la puissance électrique générée par le train considéré.

$$\begin{cases} R_d = \dfrac{\Delta V}{\Delta I} = \dfrac{E_+ - E_-}{I_{max}} \\ I_{max} = \dfrac{P_{train}}{E_-} \end{cases} \Rightarrow R_d = \frac{(E_+ - E_-) \times E_-}{P_{train}} \tag{2.8}$$

2.5 Modélisation d'une ligne de métro

2.5.1 Modélisation du système d'électrification ferrovaire

La modélisation électrique d'une ligne ferroviaire a fait l'objet de nombreux travaux, tant pour des réseaux alimentés en courant alternatif que continu. [45], [51] et [52] présentent des méthodes pour modéliser une ligne ferroviaire alimentée en courant alternatif, mais les modèles équivalents qui en résultent sont en beaucoup de points similaires aux modèles définis dans des cas d'électrification en courant continu par [46], [48], [34], [53], [54], [55] ou encore [56].

2.5.2 Présentation des hypothèses de modélisation

Pour les besoins de l'étude, la complexité du réseau électrique de ligne de métro est volontairement simplifiée, pour nous concentrer sur le fonctionnement des sous-stations d'alimentation et des trains. Ainsi, la partie amont d'acheminement de l'énergie est occultée et seul le réseau électrique de traction est étudié.

Diverses hypothèses simplificatrices sont formulées pour mettre en oeuvre le modèle électrique de la ligne afin de réduire la complexité du problème et de ne retenir que les paramètres les plus influents :

— Une sous-station est assimilée à une source de tension E_0 en série avec une résistance R_0 représentant la chute de tension interne des PR : $E_0 = 810\text{V}$; $R_0 = 30.10^{-3}\Omega$.

— Afin de limiter les chutes de tension et les tensions rail sol, une mise en parallèle des rails de même polarité est réalisée de voie à voie, à intervalle régulier le long de la ligne. Une ligne de métro est alors considérée comme un rail positif et un rail négatif entre les bornes desquelles circulent les trains.

— Les rails d'alimentation sont représentés par des impédances, dont la valeur est proportionnelle à la distance entre les éléments qu'ils interconnectent. La valeur de ces impédances de ligne varie également selon la présence de feeders sur la ligne.

— Une impédance équivalente de ligne est définie comme la somme de l'impédance des rails positif et négatif (2.9).

$$Z_{i,i+1} = R_{i,i+1} + R'_{i,i+1} \tag{2.9}$$

— En première approximation, les impédances de lignes sont considérées comme purement résistives : $Z_{eq} = R_{eq}$; $R_{i,i+1} = 53.10^{-6}\Omega.km^{-1}$; $R'_{i,i+1} = 26.10^{-6}\Omega.km^{-1}$. Le caractère inductif et capacitif de la ligne est négligé.

— La tension du rail négatif est considérée constante en tout point de la ligne et égale à 0V.

— Les courants de fuite sont considérés nuls quelles que soient les conditions de fonctionnement du réseau.

— Les sous-stations d'alimentation sont non-réversibles, ce qui implique que le courant régénéré lors des phases de freinage ne peut être renvoyé sur le réseau HTA.

— Du fait du déplacement des trains, la modélisation de la ligne de métro doit être faite en régime permanent quasi-statique ; en mettant à jour la position des trains à chaque pas de temps.

2.5.3 Application à un exemple simplifié

Le schéma électrique d'une ligne de métro simplifiée est présentée à la figure 2.17. Cette ligne fictive est composée de 3 sous-stations d'alimentation (noeuds 1,3 et 5), d'un train en traction (noeud 2) et d'un autre en freinage (noeud 4). Le train freineur est distinguable du fait de la résistance variable placée en parallèle de la source de courant afin de modéliser la possibilité de dissipation du freinage électrique.

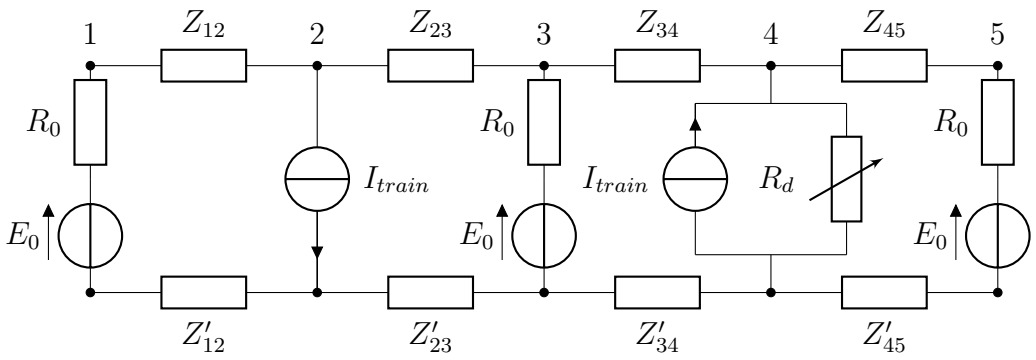


FIGURE 2.17 – Schéma électrique d'une ligne simplifiée.

2.5.4 Analyse nodale modifiée

L'analyse nodale modifiée repose sur l'utilisation des lois de Kirchoff pour déterminer les tensions nodales et les courants circulant dans les branches d'un circuit électrique.

La loi des mailles permet d'établir une relation entre les tensions au sein d'une maille d'un circuit. Cette loi stipule que dans une maille la somme algébrique des tensions est nulle. Ainsi, pour chaque maille du réseau composée de k branches , il est possible d'écrire la relation (2.10).

$$\sum_{i=1}^{k} U_i = 0 \tag{2.10}$$

La loi des noeuds permet d'établir une relation entre les courants électriques parcourant un noeud du circuit : la somme algébrique des courants entrant dans un noeud est égale à la somme de ceux sortant du même noeud. Pour un noeud constitué de k branches, l'équation (2.11) est obtenue.

$$\sum_{i=1}^{k} I_i = 0 \tag{2.11}$$

En combinant les lois de Kirchhof et la loi d'Ohm, qui relient l'intensité du courant électrique traversant un dipôle avec la tension à ses bornes, il est possible d'écrire les équations reliant les éléments électriques entre eux.

Nous obtenons alors la formulation matricielle suivante (2.12), où $[V]$ et $[I]$ sont respectivement les vecteurs des tensions et des courants nodaux et $[Y]$ est la matrice d'admittance qui synthétise la connaissance de l'état du réseau.

En effet, cette matrice renseigne sur la distance entre les trains et les sous-stations d'alimentation, mais aussi sur la réceptivité du réseau. $[Y]$ est définie comme l'inverse de la matrice d'impédance $[Z]$ correspondante.

$$[Y].[V] = [I] \quad où \quad [Y] = \frac{1}{[Z]} \tag{2.12}$$

En considérant le circuit simplifié de la figure 2.17, en exprimant les impédances par leurs admittances équivalentes et en explicitant les paramètres électriques du réseau, il est possible d'obtenir une vue encore plus synthétique de la relation électrique qui unit chaque noeud du réseau (figure 2.18).

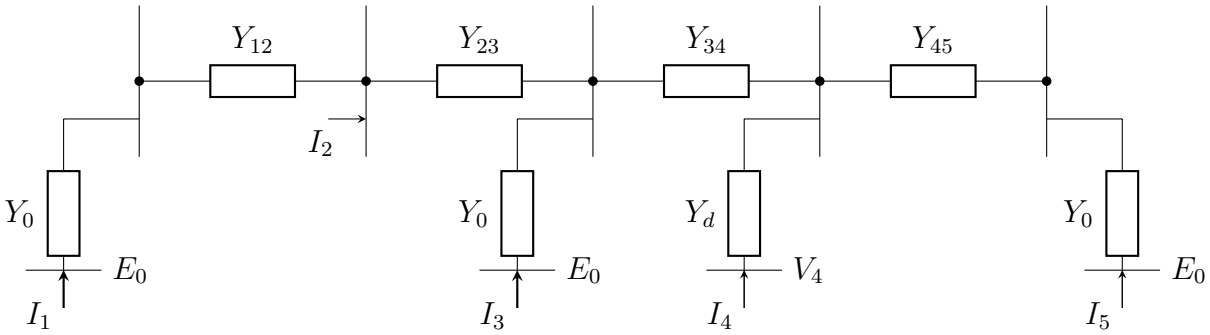


FIGURE 2.18 – Représentation du réseau électrique avec des admittances.

La construction de la matrice d'admittance du réseau est alors définie par (2.13), où l'expression de l'admittance de ligne reliant les noeuds 1 et 2 est donnée par : $Y_{12} = \dfrac{1}{Z_{12} + Z'_{12}}$.

$$Y = \begin{vmatrix} Y_{12} + Y_0 & -Y_{12} & 0 & 0 & 0 \\ -Y_{12} & Y_{12} + Y_{23} & -Y_{23} & 0 & 0 \\ 0 & -Y_{23} & Y_{23} + Y_{34} + Y_0 & -Y_{34} & 0 \\ 0 & 0 & -Y_{34} & Y_{34} + Y_{45} + Y_d & -Y_{45} \\ 0 & 0 & 0 & -Y_{45} & Y_{45} + Y_0 \end{vmatrix} \tag{2.13}$$

Cette méthode de définition de la matrice d'admittance d'un réseau ferroviaire électrique peut alors être étendue à l'étude de lignes réelles plus complexes.

La formulation exhaustive de la matrice des courants nodaux I et de celle des tensions nodales V pour le cas d'étude considéré, est rappelée en (2.14).

$$I = \begin{pmatrix} -\frac{E_0}{R_0} \\ \frac{P_2}{V_2} \\ -\frac{E_0}{R_0} \\ \frac{(E_+ - V_4).P_4}{(E_+ - E_-).E_-} \\ -\frac{E_0}{R_0} \end{pmatrix} \qquad V = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{pmatrix} \tag{2.14}$$

L'analyse nodale met en évidence un système d'équations non linéaires qu'il sera nécessaire de résoudre afin de déterminer la valeur des éléments des matrices $[V]$ et $[I]$.

2.6 Résolution d'un problème de répartition des charges

2.6.1 Analogie avec une résolution load flow

Le problème de répartition des charges est un des principaux défis qui se présentent aux gestionnaires de réseaux électriques de transport et de distribution. Ce problème a pour principaux objectifs d'évaluer le transit de puissance au sein du réseau dans un but de gestion et de plannification, mais également de déterminer l'impact sur le réseau électrique d'actions telles que l'ajout ou le retrait de charges, de sources de puissances ou de nouvelles interconnexions.

Ce problème est couramment mentionné dans la littérature par les termes anglais *load flow* ou *(optimal) power flow*.

Dans sa définition la plus simple, une étude de *load flow* consiste à déterminer, en régime stationnaire, la valeur des tensions, courants et puissances échangés dans un système électrique soumis à des conditions de charge données.

Le terme *optimal* désigne quant à lui la volonté des gestionnaires de régler les paramètres électriques et énergétiques du réseau afin d'assurer la fiabilité, la robustesse et surtout la fourniture de l'énergie aux clients finaux. En outre, dans un contexte de libéralisation du marché de l'énergie électrique, le terme *optimal* fait également référence à un besoin de minimiser les coûts d'exploitation d'un réseau électrique.

Cependant, dans notre cas d'étude, l'hypothèse sera faite que les études réalisées en amont de la construction d'une ligne de métro ont déjà permis de mettre en place un réseau électrique correctement dimensionné pour assumer les contraintes technico-économique d'exploitation.

2.6.2 Historique de la résolution de problème de load flow

L'une des premières résolutions automatisées d'un problème de répartition des charges a été réalisée par [57] en 1956 à l'aide d'une heuristique itérative utilisant une formulation par matrice d'admittance.

Une décennie plus tard, [58] passe en revue les techniques développées, par la suite, pour la résolution d'un problème de *load flow*. Les formulations du réseau par matrice d'admittance ou d'impédance sont alors majoritairement utilisées. Historiquement, outre les heuristiques itératives, la méthode de résolution dite de Gauss-Seidel a été la première employée avant que la méthode de Newton-Raphson ne la supplante à la fin des années 1960 [59].

La méthode de Gauss-Seidel a l'avantage de nécessiter peu de mémoire et de ne pas résoudre de système matriciel, cependant, elle présente une vitesse de convergence lente quand le nombre de noeuds du réseau augmente, ce qui a poussé les chercheurs à utiliser d'autres algorithmes itératifs comme la méthode de Newton-Raphson [60].

La méthode de Newton-Raphson est devenue la technique de résolution privilégiée pour les systèmes de *load flow* [61] et a été étendue, par la suite, à l'étude de systèmes ferroviaires életriques [62], [63].

L'évolution constante des moyens de calcul a ensuite vu l'essor d'un certain nombre de logiciels commerciaux permettant de modéliser puis de simuler des réseaux ferroviaires plus ou moins complexes.

— Sidytrac (Siemens)
— Elbas (Alstom)
— Energy Management Model (Université Carnegie Mellon - USA)
— ECOtranz (Bombardier)
— Multi-Train Simulator (Université de Birmingham - UK)
— Fabel (Enotrac)
— SIMTRAC (ABB & Adtranz)

Cette liste est loin d'être exhaustive, mais cette abondance d'outils montre à quel point les problématiques liées à la connaissance des flux de puissances dans un réseau ferroviaire électrique sont prises au sérieux tant par les industriels que les universitaires.

2.6.3 Méthode de Newton-Raphson

2.6.3.1 Présentation générale

La méthode de Newton-Raphson est un algorithme de résolution numérique capable de déterminer les racines d'une fonction réelle par approximations successives. Cette méthode est très utilisée du fait de ses propriétés de convergence quadratique locale. En d'autres termes, la convergence est atteinte en un nombre limité d'itérations si le point de départ de la résolution est pris assez proche de la solution finale recherchée. Si cette condition n'est pas respectée, la convergence pourra être atteinte en un nombre plus élevé d'itérations.

Bien qu'initialement prévue pour la résolution de fonction unidimmensionnelle, l'algorithme a été étendu à l'étude de fonction à dimension infinie.

Ici, cette méthode est utilisée pour calculer la ou les racines du système d'équation non linéaires (2.12). En d'autres termes, la recherche des racines revient à trouver les valeurs pour lesquelles la fonction étudiée s'annule (2.15).

$$f(x) = 0 \tag{2.15}$$

Dans sa version unidimmensionnelle, la méthode de Newton-Raphson est l'étude d'une fonction f d'une variable réelle infiniment dérivable et a, un point au voisinage duquel la fonction f est définie. Le développement limité ou série de Taylor, qui donne une approximation polynomiale de la valeur de f au point a, est donné par (2.16), où $f(a)^n$ représente la n-ième dérivée de f au point a.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \cdots \tag{2.16}$$

La méthode de Newton-Raphson se contente d'utiliser un développement limité de f à l'ordre 1. Cela revient à considérer que la fonction étudiée est quasiment égale à sa tangente en ce point.

Dans la pratique, nous partons d'un point initial x_0 appartenant à l'ensemble de définition de la fonction (2.17) ; et à chaque itération, une nouvelle solution, qui doit se rapprocher de la valeur réelle de la racine de la fonction f, est obtenue. Cela nous permet d'en déduire la relation de récurrence (2.18).

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) = 0 \tag{2.17}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.18}$$

Dans un problème à n-dimension, la formulation matricielle devient (2.19), où J est la matrice jacobienne de f et $i, j = 1, ..., n$.

$$x_{n+1} = x_n - J^{-1} f(x_n) \qquad avec \qquad J_{i,j} = \frac{\delta f_i}{\delta x_j} \tag{2.19}$$

En considérant une fonction F définie de $\mathbb{R}^n$ dans $\mathbb{R}^n$ (ici n=2) et $X = (x_1, x_2)$, une variable à 2 dimensions, l'expression de la matrice jacobienne de F est explicitée par (2.20).

$$J_F(x) = \begin{pmatrix} \frac{\delta F_1(X)}{\delta x_1} & \frac{\delta F_1(X)}{\delta x_2} \\ \frac{\delta F_2(X)}{\delta x_1} & \frac{\delta F_2(X)}{\delta x_2} \end{pmatrix} \tag{2.20}$$

2.6.3.2 Formulation mathématique du problème

L'objectif de la résolution est de déterminer les solutions de l'équation du réseau (2.21), où V_n et I_n sont respectivement les vecteurs de tensions et de courants aux noeuds du réseau et Y_n est la matrice d'admittance du réseau. La matrice jacobienne se déduit alors immédiatement (2.22).

$$F(V_n) = Y_n.V_n - I_n = 0 \tag{2.21}$$

$$F'(V) = Y_n - \frac{\delta I_n}{\delta V_n} \tag{2.22}$$

En outre, comme le montre l'expression de la matrice des courants nodaux I (2.14), le terme $\frac{\delta I_n}{\delta V_n}$ voit son expression être modifiée en fonction de l'état du train au noeud considéré. Lorsque le train est en traction ou en phase de freinage purement électrique, ce terme devient (2.23) et lorsque le train est en phase de freinage avec dissipation électrique de ce freinage (2.24).

$$\frac{\delta I}{\delta V} = -\frac{P_{trains}}{V^2} \tag{2.23}$$

$$\frac{\delta I}{\delta V} = -\frac{P_{trains}}{((E^+ - E^-) \times E^-} \tag{2.24}$$

2.6.3.3 Algorithme de résolution

L'algorithme de résolution est synthétisé dans le synoptique de l'algorithme 1. Il s'agit d'un processus itératif composé de 3 étapes : le calcul de la matrice jacobienne de la fonction à annuler, l'inversion de cette matrice puis le calcul d'une nouvelle solution. Cette relative simplicité d'implémentation est également l'un des facteurs ayant contribué à la popularisation de la méthode de Newton-Raphson.

Algorithme 1 Algorithme de Newton-Raphson

Création d'une solution initiale $V^{(0)}$
Tant que $|V^{n+1} - V^n| > \epsilon$ **Faire**
  Calcul de la matrice jacobienne J_F
  Inversion de la matrice jacobienne J_F
  Calcul de la nouvelle solution $V^n = V^{n-1} - J^{-1}F(V_{n-1})$
Fin du Tant que

La tolérance ϵ est spécifiée de sorte que la précision sur la résolution des tensions nodales soit assez fine, mais également pour que le temps de résolution reste faible. Après divers essais, la valeur $\epsilon = 10^{-3}$ a été retenue comme étant celle permettant le meilleur compromis entre précision et temps de résolution.

2.6.4 Méthode de Broyden

La méthode de Newton-Raphson présente des propriétés de convergence intéressantes, cependant, pour certaines classes de problème, des améliorations de la méthode doivent être apportées. En effet, il est nécessaire de calculer n^2 dérivées partielles à chaque itération de cette méthode.

Ainsi, lorsque la dimension du système est trop importante, le temps de calcul de la matrice jacobienne devient prohibitif. De fait, la résolution du système d'équation devient une opération très coûteuse en temps de calcul.

2.6.4.1 Mise à jour de Broyden

Pour palier à ces inconvénients, une astuce courante consiste à rechercher une approximation mathématique, où une valeur approchée de la solution du système d'équation est obtenue à chaque itération. La résolution de l'équation (2.21) étant censée converger en un nombre fini d'itérations, les matrices jacobiennes issues d'itérations consécutives doivent être *assez proches*. En théorie, il serait donc possible de définir un terme M qui vérifie la relation de récurrence $B_{n+1} = B_n + M$, où B_n est une valeur approchée de la jacobienne de la fonction. La résolution est alors dite *quasi-newton*.

En reprenant la définition de la jacobienne équivalente à l'équation d'une tangente, et en reprenant l'expression (2.19), le système (2.25) est obtenu, où B_n représente l'approximation de la jacobienne à la n-ième itération.

$$\begin{cases} B_{n+1} = \dfrac{F(X_{n+1}) - F(X_n)}{(X_{n+1} - X_n)} \\ B_n = \dfrac{F(X_n)}{(X_{n+1} - X_n)} \end{cases} \tag{2.25}$$

A la première itération, la valeur exacte de la jacobienne : $B_0 = J_F(X_0)$ est utilisée pour calculer une nouvelle solution X_1. Puis, en exprimant la différence $B_1 - B_0$ et en simplifiant son expression, les relations (2.26) et (2.27) sont obtenues.

$$B_1 - B_0 = \frac{F(X_1) - F(X_0)}{X_1 - X_0} - \frac{-F(X_0)}{X_1 - X_0} \tag{2.26}$$

$$B_1 = B_0 + \frac{\Delta F_1 - B_0.d_0}{d_0.d_0} \otimes d_0 \quad avec \quad \Delta F_1 = F(X_1) - F(X_0) \quad et \quad d_0 = -B_0^{-1}F(X_0) \tag{2.27}$$

Par généralisation, l'équation dite de Broyden est alors définie par (2.28).

$$B_{n+1} = B_n + \frac{\Delta F_n - B_n.\Delta X_n}{\|\Delta X_n\|^2} \otimes \Delta X_n \quad avec \quad \Delta X_n = X_{n+1} - X_0 \tag{2.28}$$

2.6.4.2 Mise à jour de Sherman-Morrison

Cependant, lors du processus de résolution, il nous est nécessaire de connaitre l'inverse de la matrice jacobienne. La formule de Sherman-Morrison permet alors d'obtenir

l'approximmation de l'inverse de la jacobienne (2.29), dont l'usage sera privilégié afin de limiter le coût en calcul de la résolution.

$$B_n^{-1} = B_{n-1}^{-1} + \frac{\Delta X_n - B_{n-1}^{-1}\Delta F_n}{\Delta X_n^T B_{n-1}^{-1}\Delta F_n} \otimes \Delta X_n B_{n-1}^{-1} \tag{2.29}$$

2.6.4.3 Remarques générales sur la méthode

Les méthodes de Newton-Raphson et Broyden ont de bonnes propriétés de convergence.

Ainsi, lorsque le nombre d'itérations dépasse un seuil prédéfini [8], la solution initiale V_0 doit être modifiée et la procédure de résolution est relancée, puisque la convergence de ces algorithmes dépend pour beaucoup de l'itéré initial.

De plus, il est nécessaire de vérifier les niveaux de tensions des sous-stations d'alimentation, afin de vérifier que le courant régénéré en excès ne soit pas renvoyé sur le réseau. Lorsque la tension aux bornes d'une sous-station excède une certaine valeur, celle ci doit être bloquée. En pratique, le modèle de la sous-station doit être modifié en augmentant la valeur de la résistance interne à une très haute valeur pour simuler la présence d'une diode bloquante.

2.6.5 Résolution par heuristique itérative

Dans [64], Cai présente un problème très similaire à celui défini en section 2.5.3, et y décrit une méthode de résolution remarquable qui ne fait pas appel à des approximations par dérivées successives.

La méthodologie présentée dans ces travaux a donc été reprise pour pouvoir comparer les performances de cette heuristique avec la méthode de Newton-Raphson / Broyden.

L'auteur définit tout d'abord la relation (2.30) pour modéliser les liaisons électriques entre les trains et les sous-stations présents aux noeuds du réseau continu.

$$Y_{reseau}.V_{noeud} = I_{noeud} \tag{2.30}$$

En outre, le déplacement de chaque train impose des conditions de charge, ce qui a pour effet de conditionner les échanges de puissance qui s'effectuent entre les sous-stations et les trains.

En chaque noeud du réseau où un train est présent, il est alors possible de vérifier la relation 2.31.

$$P_{trains} = V_{trains}.I_{trains} \tag{2.31}$$

La détermination des paramètres électriques en chaque noeud du réseau se fait ensuite par un procédé itératif, résumé par l'algorithme 2 [9].

La tolérance ϵ adopte la même valeur que pour l'algorithme 1.

8. Ce seuil dépend du problème considéré, il doit donc être défini par essais et erreurs en analysant le nombre moyen d'itération pour que la résolution s'effectue correctement.

9. Le lecteur intéressé, pourra se référer aux travaux de [64], pour visualiser un synoptique plus exhaustif de la méthode.

Algorithme 2 Algorithme d'heuristique itérative

1: Création d'une solution initiale $V^{(0)}$
2: **Tant que** $|V^{n+1} - V^n| > \epsilon$ **Faire**
3: Calcul du vecteur courant $I_n = Y_n.V_n$
4: $I_{trains} \leftarrow I_n$
5: Calcul du nouveau vecteur tension $V_{trains} = \frac{P_{trains}}{I_{trains}}$
6: $V_n \leftarrow V_{trains}$
7: **Fin du Tant que**

Outre sa relative simplicité, cette méthode de résolution présente l'avantage de ne pas nécessiter de manipulations de matrices coûteuses en temps de calcul.

Cependant, dans [64], de nombreuses étapes de l'heuristique ont pour objectif d'effectuer une vérification du respect des conditions électriques et énergétiques en chaque nœud du réseau, pour s'assurer que le processus de résolution converge vers une solution admissible qui conserve une réalité physique.

2.6.6 Résultats et performances de la résolution

Certains problèmes de non-convergence surviennent lors du processus résolution de Newton-Raphson. En effet, la procédure appliquée nécessite d'inverser la matrice jacobienne à chaque itération.

Or, dans certaines conditions de fonctionnement, la jacobienne présente des singularités et l'inversion de cette matrice entraine la divergence de l'algorithme de résolution. Ainsi, pour quasiment 10% des cas étudiés, la matrice jacobienne présente des singularités.

D'après l'expression de la jacobienne (2.22), les singularités sont générées soit par l'expression de la matrice d'admittance, soit par la dérivée des courants nodaux par rapport aux tensions nodales.

Une étude attentive des conditions de singularité indique que dans une très grande majorité des cas, l'expression de la matrice d'admittance en est la cause.

En effet, lorsque la distance entre les trains et les sous-stations d'alimentation est faible, la valeur de l'admittance au noeud considéré devient très grande, ce qui provoque une non-convergence lors de l'exécution de l'algorithme de résolution.

En revanche, en utilisant la méthode de Broyden pour mettre à jour la valeur de la matrice d'impédance, les problèmes de non-convergence disparaissent d'eux même puisque l'étape qui engendrait la singularité de la matrice jacobienne est supprimée et le temps de calcul est également diminué.

Une comparaison de vitesse de résolution entre la méthode de Newton-Raphson et la méthode de Broyden, pour des cas ne présentant pas de singularités, montre que la méthode Quasi-newton est en moyenne 17% plus rapide en terme de durée de résolution.

Une analyse comparative des performances de convergence de la méthode Quasi-Newton de Broyden et de l'heuristique développée par Cai est effectuée. Deux critères sont analysés : le temps de convergence et le nombre d'itérations pour y arriver. La méthode de Newton-Raphson n'est pas considérée puisqu'elle ne converge pas pour tous les cas d'études.

Pour cela, la ligne de Turin a été prise comme référence, avec 16 trains en exploitation sur un tour de boucle entier ; chaque intervalle possible a été simulé et les paramètres électriques nodaux de la ligne à chaque pas de temps ont été déterminés par les deux méthodes de résolution. La résolution a été effectuée avec un pas d'échantillonnage de 1s.

Chaque simulation nécessite en moyenne 3060 évaluations, et 12 simulations ont été nécessaires pour étudier l'ensemble des intervalles possibles avec un carrousel de 16 trains.

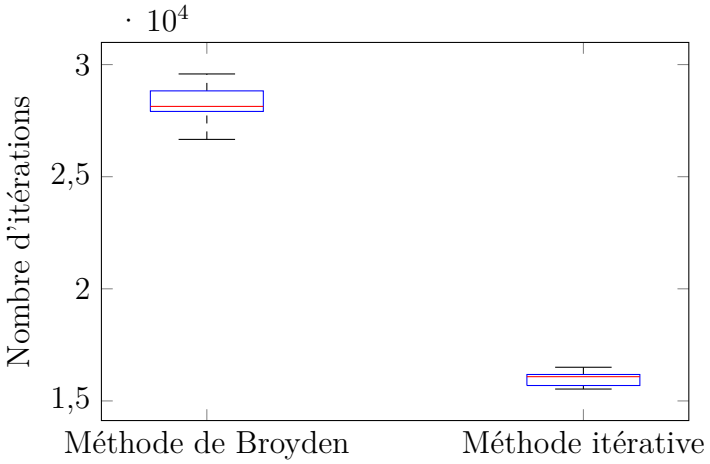


FIGURE 2.19 – Distribution statistique du nombre d'itérations pour effectuer la résolution.

La figure 2.19 montre que la méthode de Broyden nécessite en moyenne un plus grand nombre d'itérations que l'heuristique itérative pour solutionner chacunes des 12 simulations, cependant, le temps de calcul par simulation est plus faible (figure 2.20). La méthode de Broyden met en moyenne 18s / 28000 itérations pour déterminer les paramètres électriques nodaux des 3060 pas de temps de simulation tandis que l'heuristique itérative formulée par Cai met en moyenne 19s / 16000 itérations pour réaliser la même tâche.

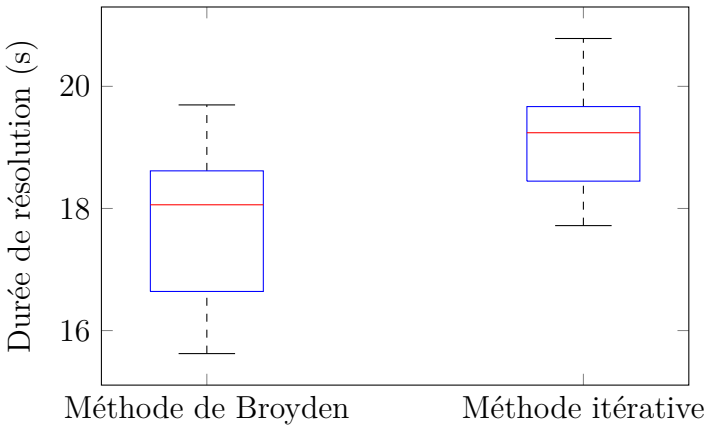


FIGURE 2.20 – Distribution statistique de la durée de résolution.

Une résolution par la méthode Quasi-Newton de Broyden sera donc privilégiée pour déterminer les paramètres électriques du réseau électrique, du fait de sa rapidité de convergence en terme de durée de résolution.

2.7 Conclusion

Dans ce chapitre, une brève introduction sur les systèmes VAL et Néoval a été effectuée, puis les principaux éléments constitutifs d'une ligne de métro automatique ont été présentés en détails : le réseau de traction et le matériel roulant. Des modèles mécaniques et électriques de ces éléments ont été établis et des hypothèses simplificatrices ont été formulées afin d'alléger la modélisation de la ligne de métro sans en affecter la précision. Ces différents modèles ont alors été agrégés pour simuler une ligne de métro générique.

Des essais réalisés sur la ligne de métro de Turin ont permis de mettre en évidence le lien entre l'évolution de la tension locale perçue par le train en fonction de la puissance électrique de freinage renvoyée sur la ligne.

Ensuite, trois méthodes de résolution itératives ont été introduites et deux de ces méthodes ont été retenues afin de déterminer les paramètres électriques des sous-stations et des trains à chaque instant. Puis, ces techniques itératives ont été mises en oeuvre sous le logiciel Matlab et leurs performances de convergence ont été analysées sur un cas d'étude.

Cette étude comparative a permis de choisir la méthode Quasi-Newton de Broyden comme technique de résolution des paramètres nodaux d'un réseau électrique.

Après avoir modélisé une ligne de métro et avoir introduit une méthode de résolution capable de déterminer les paramètres électriques de la ligne, la suite des travaux va se focaliser sur la mise en œuvre de méthodes d'optimisation visant à réduire la consommation d'énergie au sein d'une ligne de métro.

Chapitre 3

Optimisation des paramètres d'exploitation

« Il y a toujours quelques individus que le hasard isole, ou que la génétique favorise. »

Ronald Wright

Sommaire

3.1 Introduction

Dans cette thèse, le choix a été fait de se focaliser sur la récupération de l'énergie issue du freinage électrique comme moyen de réduire la consommation énergétique d'une ligne de métro automatique.

Le chapitre précédent a permis de développer un modèle énergétique d'une ligne de métro ainsi qu'une méthode de résolution pour calculer les flux de puissance qui s'opèrent au sein du réseau électrique lors du déplacement des trains.

Une analyse énergétique de l'impact des paramètres d'exploitation sur la consommation énergétique d'un carrousel peut alors être menée à la lumière de l'étude effectuée au chapitre précédent.

Ainsi, après avoir proposé une formulation du problème d'optimisation à résoudre permettant d'introduire les variables, les contraintes et la fonction objectif du problème, la première partie de ce chapitre est consacrée à l'étude de l'influence des paramètres d'exploitation sur la consommation énergétique des carrousels.

De cette étude, deux paramètres d'exploitation ressortent : l'intervalle d'exploitation et les temps d'arrêt en station.

Une méthodologie pour optimiser l'intervalle d'exploitation est ensuite définie et une analyse de la répartition des pertes énergétiques dans une ligne est effectuée. Cette analyse permet alors de fournir une première estimation des gains énergétiques potentiels.

La partie suivante de ce chapitre est dédiée au problème d'optimisation des temps d'arrêt en station : les choix de l'utilisation de méthodes issues de l'intelligence artificielle au détriment d'une approche mathématique classique sont justifiés, puis deux de ces méthodes d'optimisation sont décrites et ensuite mises en œuvre. Une discussion est alors menée pour améliorer les propriétés de convergence de ces méthodes algorithmiques.

Enfin, ces méthodes d'optimisation sont employées pour effectuer l'optimisation énergétique de tables horaires journalières. Les limites de l'optimisation hors-ligne des paramètres d'exploitation en vue de réaliser une optimisation temps réel de la consommation énergétique sont alors soulignées.

3.2 Formulation du problème d'optimisation

3.2.1 Cahier des charges

Le cahier des charges de l'optimisation énergétique hors-ligne est résumé dans le tableau 3.1 et présente les objectifs et les contraintes de l'étude, les moyens d'actions pour réaliser l'optimisation ainsi que les indicateurs utilisés pour évaluer le niveau d'atteinte de l'objectif.

Ce tableau offre un aperçu macroscopique de la problématique traitée dans ce chapitre. Son principal intérêt est de permettre d'identifier aisément les entrées, sorties et variables du problème.

Objectifs	• Optimisation énergétique de tables horaires pré-existantes
Contraintes	• Nombre de trains en ligne • Marge de variation des temps d'arrêt en station
Moyens d'action	• Sélection de l'intervalle d'exploitation le plus favorable • Sélection de la combinaison de temps d'arrêt en station la plus favorable
Indicateurs	• Taux de réutilisation du freinage électrique

TABLEAU 3.1 – Cahier des charges de l'optimisation énergétique hors-ligne d'une ligne de métro automatique

3.2.2 Sélection des paramètres les plus influents en hors-ligne

Dans la littérature, les travaux portant sur l'optimisation de la consommation énergétique de lignes ferroviaires se distinguent par l'usage de différentes variables d'optimisation :

Les profils de vitesse commerciale

Certaines études redéfinissent les profils de vitesse commercial par logique floue [65], [66], par algorithme génétique [67] ou par la méthode du simplexe [29] (en modifiant uniquement les vitesses maximales admissibles par portion d'interstation).

Tandis que d'autres choisissent d'insérer des phases de marche sur l'erre dans les parcours interstation et de déterminer leurs durées par un algorithme génétique [68], [69],[70], [71], [67], [31] , par des méthodes directes (golden section, simplexe et fibonacci) [72], ou par de méthodes indirectes de type descente du gradient [72].

Les temps d'arrêt en station et le temps de battement

— modification de la durée d'arrêt : Le temps de battement est utilisé pour moduler la durée des arrêts en station, soit par algorithme génétique [34], soit par heuristique [40], ou par logique floue [73].

— modification de l'horaire de départ : [33] propose quant à lui de modifier les horaires de départ des trains pour synchroniser les phases d'accélération et de freinage via un solveur industriel (CPLEX) et une heuristique. [74] adopte une démarche sensiblement différente en modifiant uniquement l'horaire de départ en première station et utilise pour cela une formulation mathématique résolue par programmation linéaire.

— modification de la durée des interstation : [30] utilise un algorithme génétique pour distribuer le temps de battement disponible afin d'allonger la durée des parcours interstation.

L'intervalle d'exploitation

L'étude de l'intervalle d'exploitation constitue une étape dimensionnante indispensable, réalisée par l'exploitant de la ligne ferroviaire afin d'assurer une qualité de service tout en minimisant la consommation énergétique de la ligne.

Dans le cas des lignes de métro automatique de type VAL et Néoval, les trains roulent selon des profils de vitesse pré-définis (figures 2.10, 2.11 et 2.12). La modification de la vitesse d'exploitation ne sera donc pas étudiée dans ce manuscrit. En outre, dans

notre cas d'étude, le temps de battement est largement employé pour absorber les perturbations de trafic. Cependant, une partie de cette marge de régulation peut être utilisée pour augmenter les temps d'arrêt en station afin de synchroniser les phases d'accélération et de freinage.

Ainsi, l'optimisation énergétique se concentre sur la gestion de l'intervalle d'exploitation et la modulation des temps d'arrêt en station.

3.2.3 Définition des variables utilisées

Chaque train du carrousel parcourt les $N_{station}$ stations de la ligne de métro, en suivant des profils de vitesse entre chaque station et en effectuant des arrêts en station d'une durée $s_{i,j}$ (où $i = [1, ..., N_{station}]$ et $j = [1, ..., N_{trains}]$). Pour les besoins de l'éude, nous considèrons un domaine temporel discret avec un pas d'échantillonnage variable selon le niveau de détail requis.

Dans [36], [35] et [75] les auteurs adoptent différentes variables d'optimisation pour atteindre des objectifs très similaires à ceux de notre étude : les temps d'arrivée et de départ en station ainsi que les temps de début des phases d'accélération et de freinage.

Cependant, l'utilisation de ces variables est assez délicate à mettre en oeuvre, puisque d'une part, du fait de la topographie de la ligne étudiée, les parcours interstation peuvent comporter plusieurs phases de freinage et d'accélération, d'autre part, des profils type de vitesse sont utilisés pour simuler le déplacement des trains et enfin, un tel nombre de variables d'optimisation augmente la complexité du problème.

Dans notre approche hors-ligne de l'optimisation énergétique, le nombre de variables d'optimisation est grandement réduit puisque seule la durée des arrêts en station est utilisée pour caractériser le parcours type d'un train sur la ligne.

3.2.4 Définition des contraintes

Le temps de parcours T_{p_j} est calculé comme la somme des temps de parcours interstation t_i additionnée de la somme des temps d'arrêt en station $s_{i,j}$ et du temps de battement t_{bj} (3.1). Ce temps correspond à la durée que met un train pour effectuer un tour complet de la ligne et revenir à son point de départ initial.

$$T_{p_j} = \sum_{i=1}^{N_{station}} s_{i,j} + t_i \qquad \forall j \in [\![1, N_{trains}]\!], \forall i \in [\![1, N_{station}]\!] \tag{3.1}$$

Le temps de parcours est contraint de sorte que les trains respectent la table horaire nominale pour assurer la qualité de service souhaitée par l'exploitant. La durée des parcours interstation étant fixe, il est donc nécessaire d'imposer une contrainte sur la plage de variation des temps d'arrêt en station (3.18), où $s_{i,nom}$ représente le temps d'arrêt nominal que doit effectuer un train à la station i, tandis que les paramètres Δs_{1i} et Δs_{2i} permettent de modifier l'amplitude de la modulation.

$$s_{i,nom} - \Delta s_{1i} \leq s_{i,j} \leq s_{i,nom} + \Delta s_{2i} \tag{3.2}$$

La valeur de ces paramètres est généralement imposée par le besoin d'assurer le transit des utilisateurs sur la ligne : une fois arrivé en station, le temps d'arrêt doit

permettre la descente et la montée des passagers. Ainsi, la marge d'évolution du paramètre Δs_{1i} est assez faible, en première approximation, on prendra $\Delta s_{1i} = 10\% \ \ s_{i,nom}$.

En outre, le temps de parcours total des trains ne doit pas excéder une valeur spécifiée afin ne pas biaiser la régulation de trafic.

L'intervalle d'exploitation I est également contraint pour assurer les exigences en matière de sécurité et de qualité de service (3.3).

$$\frac{T_p}{N_{trains}} \leq I \leq \frac{T_p}{N_{trains} - 1} \tag{3.3}$$

Le paramètre Δs_{2i} devra alors respecter la contrainte (3.4), afin que la contrainte (3.3) soit toujours respectée.

$$\sum_{i=1}^{N_{station}} (\Delta s_{2i}) \leq T_p - I * N_{trains} \tag{3.4}$$

Il est intéressant de constater que la partie droite de la contrainte (3.4) correspond à la définition du battement t_{bj}, soit la marge de manœuvre temporelle dont dispose un train pour effectuer son tour de boucle.

3.2.5 Définition de la fonction objectif

Soit $P_j(t)$, la puissance électrique consommée/générée par le train j à l'instant t. La puissance globale consommée par le carrousel à l'instant t est donné par (3.5). L'énergie consommée par le carrousel sur la durée de simulation est alors définie par (3.6).

$$P_{tot}(t) = \sum_{j=1}^{N_{trains}} P_j(t) \tag{3.5}$$

$$E_{tot} = \sum_{t=0}^{T_p} P_{tot}(t) \tag{3.6}$$

Dans [75], l'auteur propose d'analyser l'efficacité énergétique d'une table horaire selon différents critères : la consommation électrique globale du carrousel E_{tot} (3.6), la puissance maximale fournie par le réseau de traction P_{max} (3.7), ou encore le coût de dépassement de la puissance souscrite Σ (3.8) (où k représente le coût du dépassement (en €.kWh^{-1}) et $P_{souscrite}$ la puissance souscrite sur la période considérée). Ces deux premiers critères sont classiquement employés dans les travaux d'optimisation de la consommation de systèmes ferroviaires comme dans [30],[33],[74] ou [31].

$$P_{max} = \max_{0 \leq t \leq T_p} P_{tot}(t) \tag{3.7}$$

$$\Sigma = \int_0^{T_p} k(t) \Big(P_{tot}(t) - P_{souscrite}(t) \Big) \, \mathrm{d}t \tag{3.8}$$

En revanche, la prise en compte du dépassement de puissance souscrite est une originalité de [75] et permet d'intégrer un critère économique supplémentaire pour diminuer la facture d'électricité payée par l'exploitant de la ligne. Bien que la définition de Σ impose d'utiliser une somme continue, en pratique, l'usage d'une somme discrète est requise du fait de la disponibilité épisodique des informations concernant la puissance globale consommée par la ligne.

Dans notre étude, hypothèse a été faite que le réseau de traction a été préalablement correctement dimensionné afin que celui-ci soit en mesure d'assurer la fourniture d'énergie aux trains même dans l'éventualité de la perte d'une sous station d'alimentation et par extension, il est considéré que le choix de la puissance souscrite est laissé à l'appréciation de l'exploitant ; de fait, les problématiques de dépassement de puissance souscrite et de lissage de la consommation ne sont pas traités dans ces travaux de thèse.

Dans la suite des travaux, seul le critère portant sur la consommation globale d'énergie E_{tot} est étudié. La fonction objectif est alors donnée par (3.9) et a pour unique but de minimiser la consommation globale d'énergie.

$$F_{objectif} = \min_{0 \leq t \leq T_p} E_{tot}(t) \tag{3.9}$$

Il est à noter que d'autres travaux comme ceux de [35] et [36] se donnent comme objectif de maximiser le temps de synchronisation entre phases d'accélération et phases de freinage, en considérant des paires de trains.

Pourtant, comme il a été montré au chapitre précédent, du fait du déplacement des trains et des spécificités propres aux réseaux DC, l'utilisation de ce critère semble être une approche trop simpliste.

Pour illustrer ces propos, prenons une ligne de métro quelconque, où dans une zone assez localisée deux trains sont en phase de freinage et un train est en phase de traction, de sorte que les phases de freinage et d'accélération se déroulent en même temps. Dans ces conditions, bien que la synchronisation des phases soit optimale, le taux de récupération de l'énergie issue du freinage sera assez faible.

En effet, localement la production étant supérieure à la consommation, le potentiel des trains freineurs va augmenter jusqu'à atteindre la tension de conjugaison du freinage, et entraînera la dissipation d'une part du freinage électrique.

3.3 Optimisation de l'intervalle

Une table horaire est définie de manière à permettre un transport fluide des usagers de la ligne. Généralement, le nombre de trains mis en service est proportionnel à l'affluence des passagers sur la ligne. Comme indiqué par la relation (3.3), à un nombre donné de trains en service, correspond une plage de variation de l'intervalle d'exploitation.

La première étape dans la conception de tables horaires consiste donc à déterminer pour chaque carrousel de trains, l'intervalle d'exploitation permettant de minimiser la consommation énergétique globale.

3.3.1 Simulation d'un carrousel établi

L'algorithme 3 présente synthétiquement la démarche mise en oeuvre pour simuler le parcours type d'un train sur un tour de boucle. Le parcours est réalisé à vitesse commerciale nominale, en effectuant des temps d'arrêt en station nominaux et en considérant qu'à l'arrêt un train consomme une puissance P_{aux}.

Algorithme 3 Simulation du parcours type d'un train sur un tour de boucle

Entrée(s) temps.arret et profil.interstation

1: **Pour** i = 1 à $N_{station}$ **Faire**
2: profil = [P_{aux} · longueur(temps.arret(i)) ; profil.interstation(i)]
3: parcours.type = [parcours.type ; profil]
4: **Fin du Pour**

Sortie(s) parcours.type

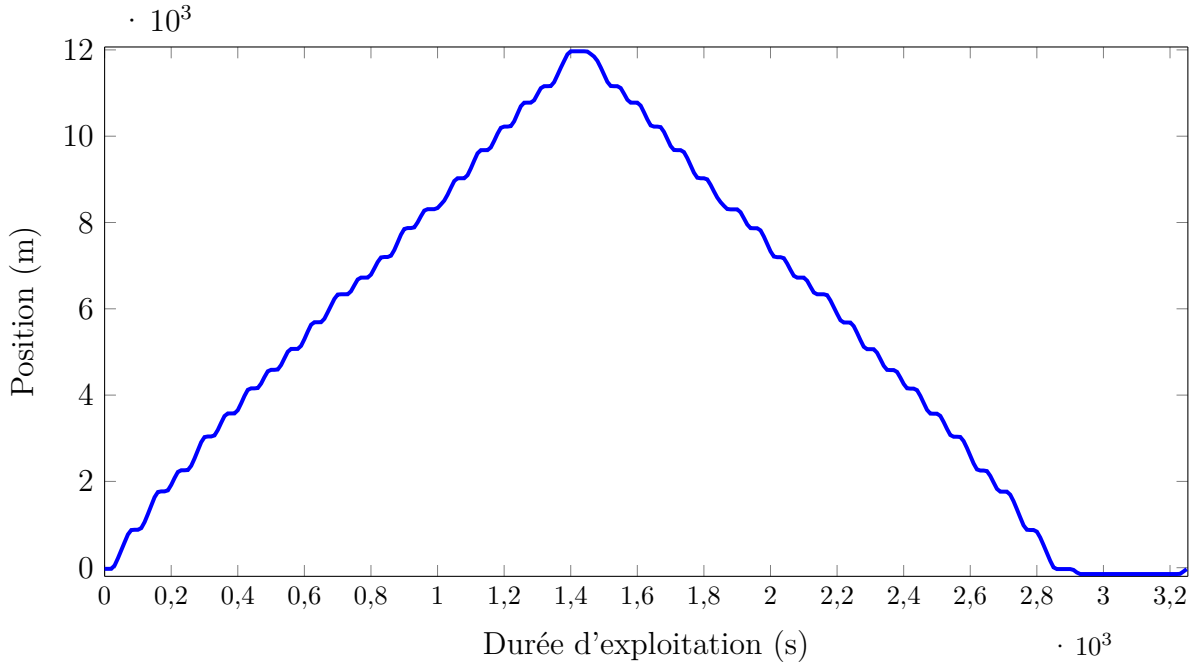


FIGURE 3.1 – Evolution de position d'un train sur un tour de boucle.

La figure 3.1 est obtenue en appliquant l'algorithme 3 à la position type d'un train sur la ligne. Cette figure est constituée d'une succession de parcours interstation et d'arrêt en station. Le train part de la première station, puis arrive en bout de ligne et repart dans l'autre sens. La phase stationnaire après le retour en première station correspond au temps de battement.

Ce parcours type est ensuite utilisé pour simuler le déplacement de l'ensemble des trains composant le carrousel. L'algorithme 4 permet de construire le parcours d'un carrousel établi sur un tour de boucle dans un cas d'exploitation sans perturbation.

La figure 3.2 présente le profil de position d'un carrousel composé de 6 trains. L'espacement temporel entre le profil de position de chaque train correspond à l'intervalle d'exploitation.

Algorithme 4 Simulation du parcours type d'un carrousel établi sur un tour de boucle de la ligne de Turin

Entrée(s) parcours.type

parcours.carrousel(: , 1) = parcours.type
Choisir un intervalle I avec l'équation (3.3)
Pour j = 2 : N_{trains} **Faire**
  Pour k = 1 : longueur(I) **Faire**
    parcours.carrousel(:,j) = *permut_circ* (parcours.type,$(j-1) \cdot I(k)$)
  Fin du Pour
Fin du Pour

Sortie(s) parcours.carrousel

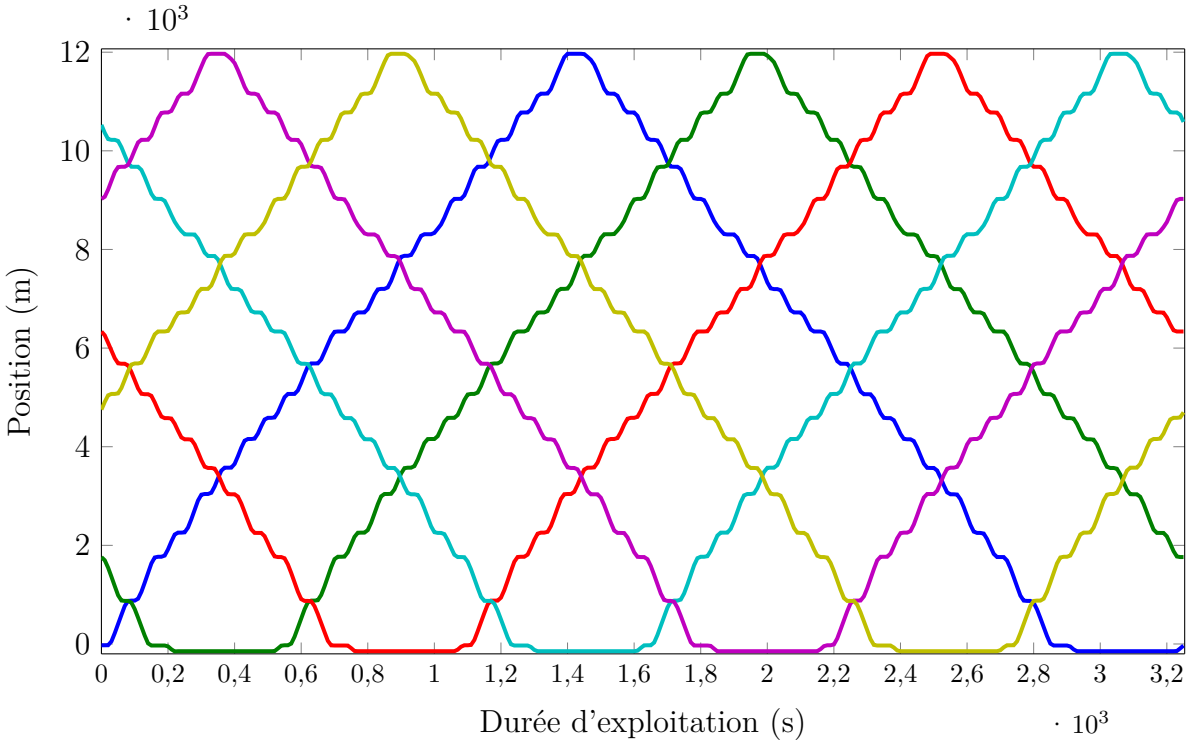


FIGURE 3.2 – Evolution de position d'un train sur un tour de boucle.

3.3.2 Etude des intervalles d'exploitation

La manière la plus efficace pour étudier l'impact de l'intervalle sur la consommation énergétique du carrousel consiste à simuler l'ensemble des intervalles d'exploitation possibles. Au chapitre précédent, il a été montré que la méthode de résolution choisie permet de calculer la consommation énergétique d'un carrousel sur un tour de boucle en environ 18s, ce qui rend cette étude exhaustive possible, sans nécessiter d'allouer des ressources de calcul supplémentaires.

La méthodologie d'optimisation de l'intervalle d'exploitation revient alors à simuler le fonctionnement d'un carrousel établi cadencé à un intervalle donné et sur un tour de boucle complet.

Les résultats de l'étude des intervalles d'exploitation sont explicités par les figures 3.3, 3.4 et 3.5. La figure 3.3 montre l'évolution de la consommation énergétique globale d'un carrousel tandis que la figure 3.4 montre l'évolution de la quantité d'énergie

générée lors du freinage qui a dû être dissipée.

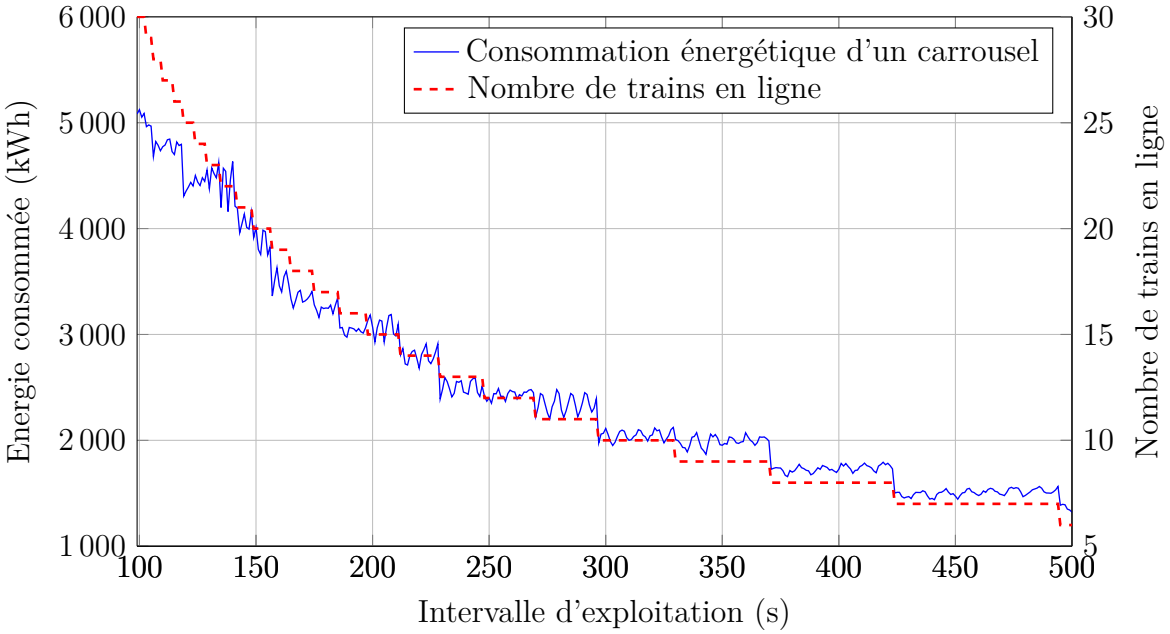


FIGURE 3.3 – Evolution de la consommation énergétique d'un carrousel en fonction de l'intervalle d'exploitation.

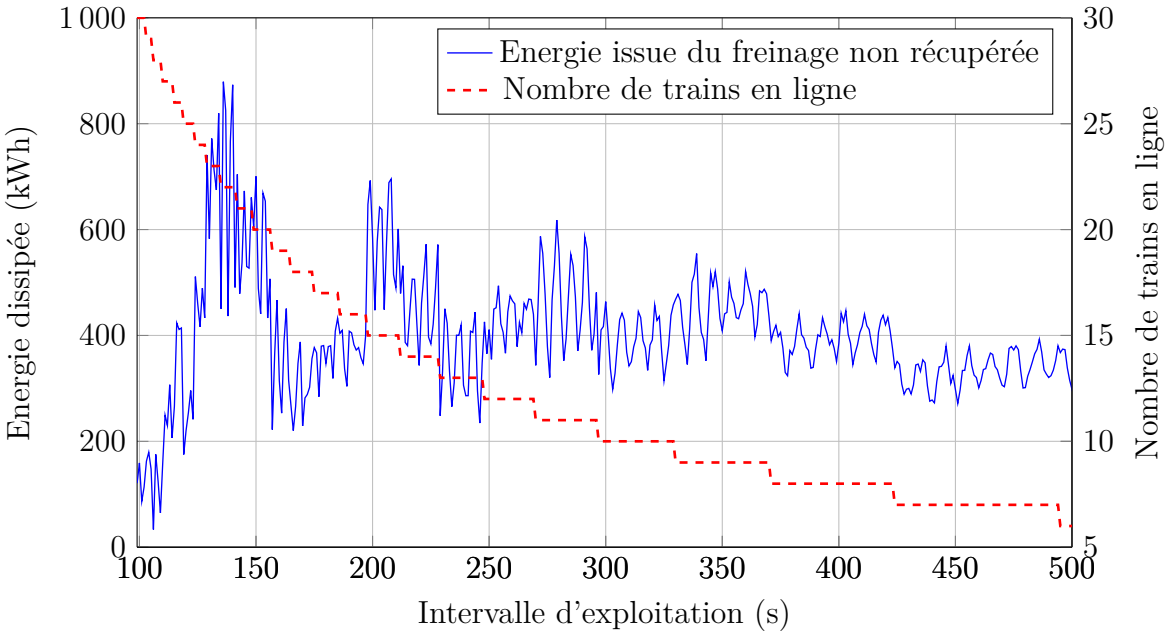


FIGURE 3.4 – Evolution de la quantité d'énergie issue du freinage non récupérée en fonction de l'intervalle d'exploitation.

Enfin la figure 3.5 synthètise les deux précédentes illustrations en exprimant l'évolution du taux de dissipation du freinage électrique par rapport à l'énergie consommée pour la traction. Chaque point de la figure 3.5 correspond alors au ratio ΔR (3.10).

$$\Delta R = \frac{E_{dissipée}}{E_{tot}} \tag{3.10}$$

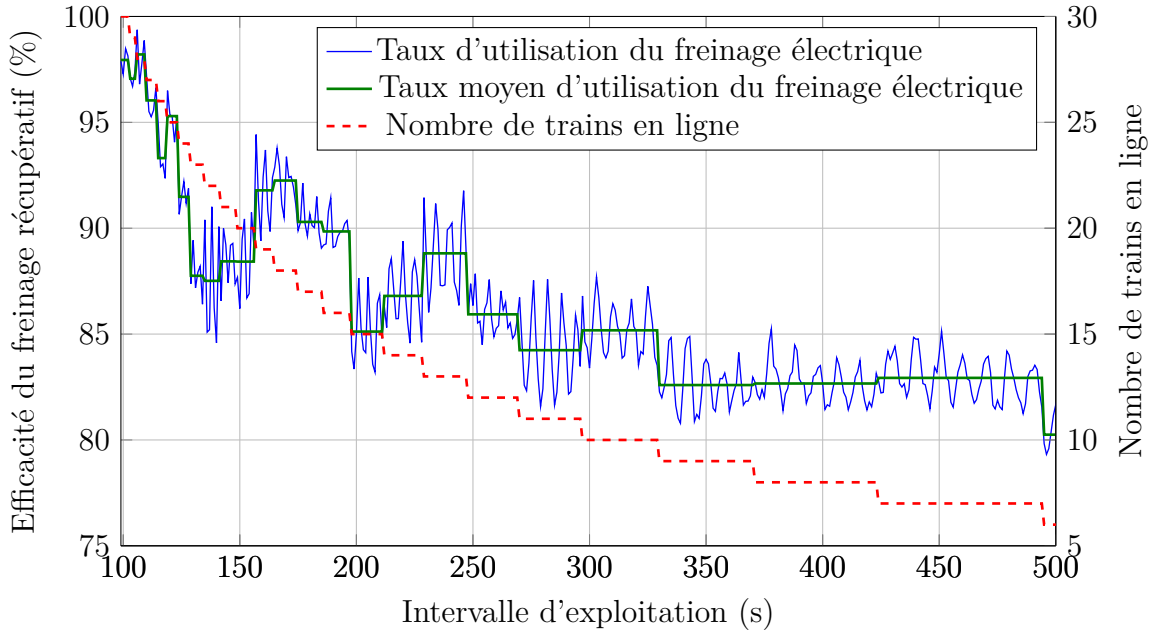


FIGURE 3.5 – Evolution du taux de récupération de l'énergie du freinage en fonction de l'intervalle d'exploitation.

Le taux d'énergie dissipée augmente lorsque le nombre de trains en exploitation diminue, en effet, moins il y a de trains en service et moins il y a de trains susceptibles de consommer l'énergie électrique générée au freinage. De plus, du fait de la configuration de la ligne, certains carrousels semblent être naturellement plus propices à une meilleure récupération du freinage électrique, comme les carrousels de 13 et 18 trains.

La figure 3.5 expose donc une estimation des gains d'énergie réalisable en sélectionnant l'intervalle le plus favorable, en considérant que l'optimum du problème serait un cas d'exploitation où la totalité de l'énergie issue du freinage électrique serait récupérée.

Sur la figure 3.4, il est possible de remarquer l'occurence régulière d'intervalles défavorables pour la récupération du freinage électrique. Cette occurrence est dépendante de l'architecture de chaque ligne de métro et est définie par l'équation empirique (3.11) [76]. Pour la ligne considérée dans le cas d'étude, les intervalles défavorables se répètent avec une période de 70s.

$$O_d = \frac{T_b}{N_{station}} \tag{3.11}$$

3.3.3 Répartition des pertes dans une ligne de métro

L'application de la méthode de résolution développée au chapitre précédent à l'étude des intervalles d'exploitation permet non seulement de déterminer la quantité d'énergie consommée par les trains, mais également la quantité d'énergie dissipée lors des périodes d'exploitation et de fait d'avoir une estimation des pertes énergétiques de la ligne.

Dans une ligne de métro classique, les pertes énergétiques sont principalement de trois types :

— Les pertes par dissipation qui correspondent à l'énergie électrique issue du freinage qui n'a pu être restituée sur le réseau et a dû être dissipée dans les rhéostats

de freinage ou par freinage mécanique.

— Les pertes en ligne engendrées par le transit de la puissance sur le réseau électrique. Il s'agit de l'énergie dissipée dans les impédances de ligne et dans une moindre mesure dans les sous-stations d'alimentation.

— Les pertes dues à la consommation des systèmes auxiliaires comme les équipements électriques en station (ascenceurs, éclairage, ventilation...) et embarqués (éclairage, ventilation, équipements de sécurité,...).

La consommation des auxiliaires étant une nécessité pour assurer le confort et la sécurité des utilisateurs, l'étude de la consommation liée à ces systèmes électriques n'est pas considérée comme un axe de recherche durant cette thèse.

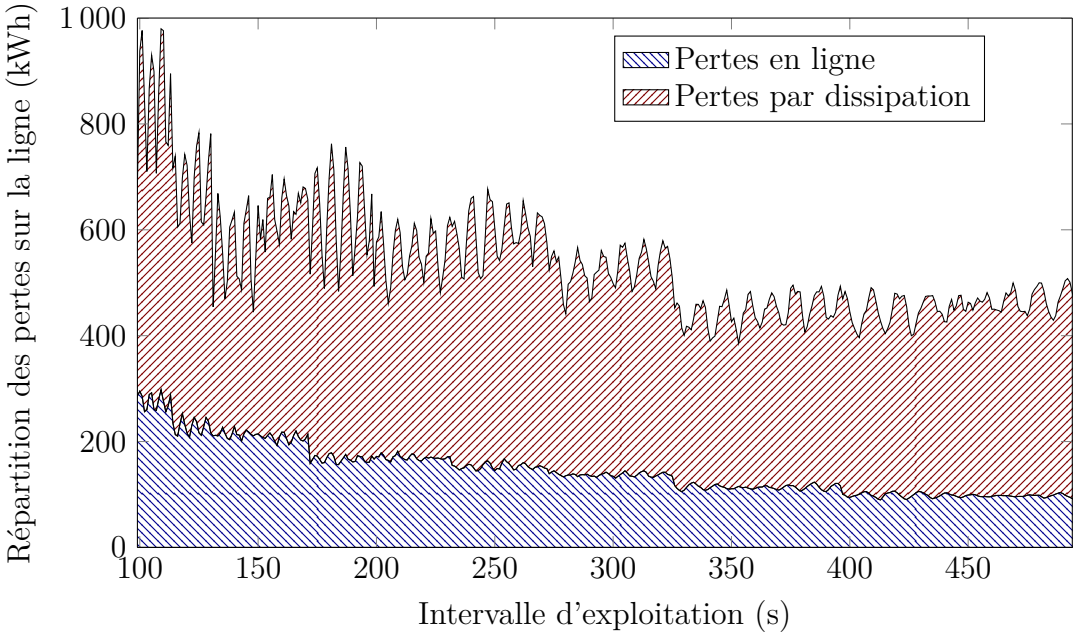


FIGURE 3.6 – Répartition absolue des pertes sans la ligne.

La figure 3.6 représente la répartition des pertes en valeur absolue en fonction de l'intervalle d'exploitation, tandis que la figure 3.7 exprime la répartition relative des pertes par rapport à l'énergie nécessaire à la traction.

Les pertes en ligne sont des pertes nécessaires sur lesquelles on ne peut influer. Le seul moyen de réduire ces pertes serait de réduire le nombre des trains en ligne et donc l'offre de service, ce qui n'est pas une option envisageable dans un contexte industriel. Sur la figure 3.7, les pertes en ligne semblent quasiment constantes et représentent environ 7% de l'énergie nécessaire à la traction. En revanche, les pertes par dissipation semblent beaucoup plus dépendantes de l'intervalle d'exploitation. Ces dernières constituent donc le critère sur lequel l'optimisation sera axé ; en effet, une gestion *intelligente* des phases de marche des trains permettrait d'augmenter le taux de récupération du freinage électrique et donc de diminuer la consommation électrique de la ligne de métro.

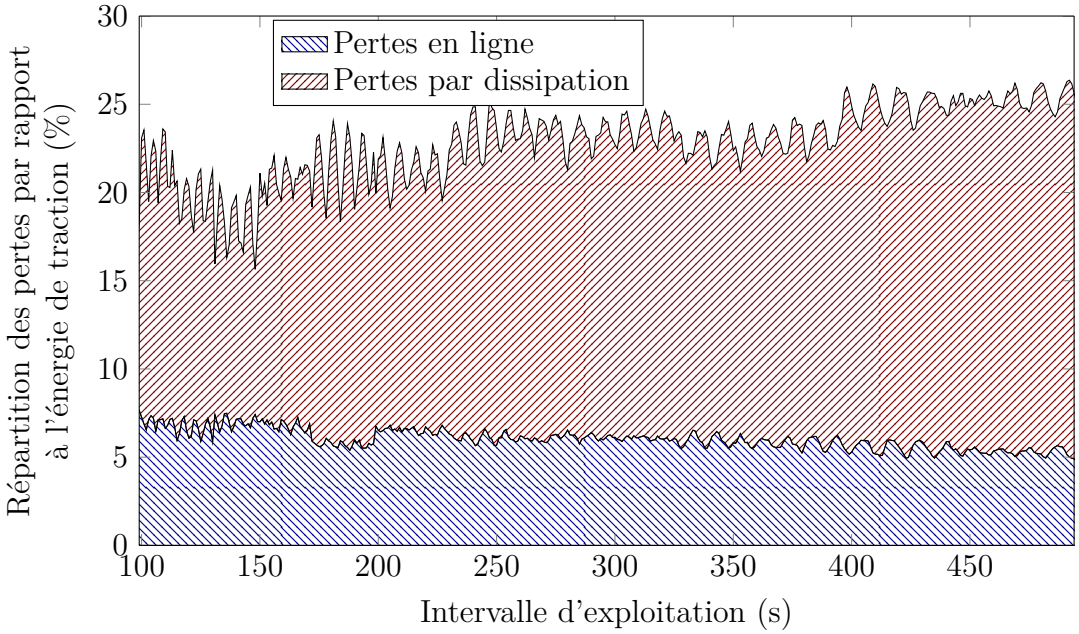


FIGURE 3.7 – Répartition relative des pertes dans la ligne par rapport à l'énergie nécessaire à la traction.

3.3.4 Analyse d'une table horaire type

Une étude statistique sur les tables horaires quotidiennes utilisées pour l'exploitation de la ligne de Turin permet de visualiser les carrousels les plus récurrents.

Le tableau 3.2 présente les caractéristiques des tables horaires utilisées pour une semaine type. Cette distribution ne prend en compte que la durée d'exploitation des phases établies. En effet, pour chaque table horaire, l'enchaînement des carrousels est différent, ce qui entraîne un grand nombre de phases transitoires distinctes à optimiser.

	Fréquence relative (%)			
Nombre de trains	Table horaire 1	Table horaire 2	Table horaire 3	Table horaire 4
8	4	20	22	24
10	7	7	5	-
12	-	-	25	39
14	6	6	3	
16	24	24	36	37
18	27	27	10	
20	19	3	-	-
25	13	13	-	-

TABLEAU 3.2 – Distribution de la fréquence d'utilisation des carrousels.

D'après la figure 3.8, les carrousels de 16, 8, 12 et 18 trains expliquent 82% de la durée d'exploitation hebdomadaire de la ligne de Turin.

En corrélant cette étude statistique aux données issues de l'optimisation de l'intervalle d'exploitation, un gain énergétique hebdomadaire d'environ 12% de la consommation nominale est théoriquement réalisable sous l'hypothèse d'une gestion optimale

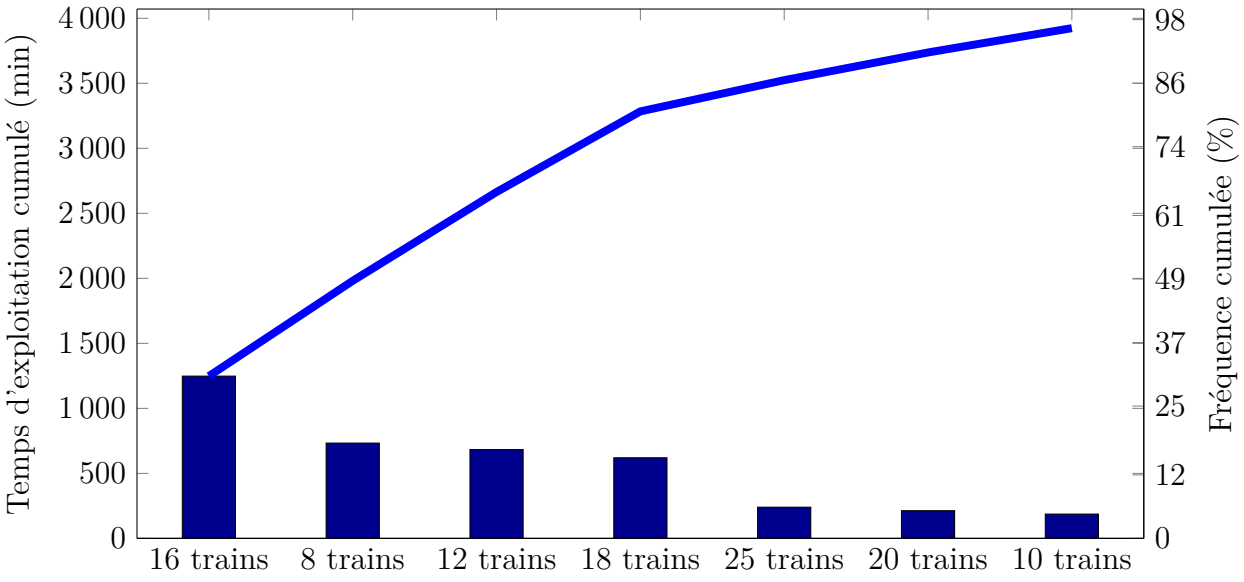


FIGURE 3.8 – Distribution de la durée d'exploitation des carrousels sur une semaine type

permettant un taux de récupération du freinage électrique de 100%. Pour notre cas d'étude, cette valeur représente ainsi le gain maximal réalisable.

3.3.5 Confrontation avec des essais sites

Les résultats des simulations précédentes ont été confrontés avec des essais réalisés sur la ligne de métro de Turin. Ces essais avaient pour objectif d'évaluer la capacité des modèles à estimer la consommation moyenne d'un carrousel sur un tour de boucle.

Seuls quelques intervalles ont pu être testés étant donné que deux tours de boucles sont nécessaires pour stabiliser la régulation d'un carrousel et ainsi obtenir une mesure fiable de la consommation énergétique.

La figure 3.9 compare la puissance moyenne qui a été mesurée pour différents intervalles et l'estimation donnée par simulation. L'utilisation des modèles présentés au chapitre précédent entraîne ainsi une erreur moyenne d'environ 6%.

Les différences constatées entre la théorie et l'expérimentation peuvent être imputées à la fois aux hypothèses simplificatrices utilisées dans les modélisations, à savoir les profils de vitesse supposés immuables et la simplicité du modèle de renvoi de puissance entre les trains, mais aussi par les aléas d'exploitations qui sont survenus lors des essais sur site et qui n'ont pas été pris en compte dans les simulations.

De plus, la comparaison concerne la valeur moyenne des puissances enregistrées lors des essais, ce qui constitue une valeur moins fiable que la quantité d'énergie consommée, puisque dans le cadre des essais, les mesures ont été effectuées avec un pas d'échantillonnage de 15 minutes.

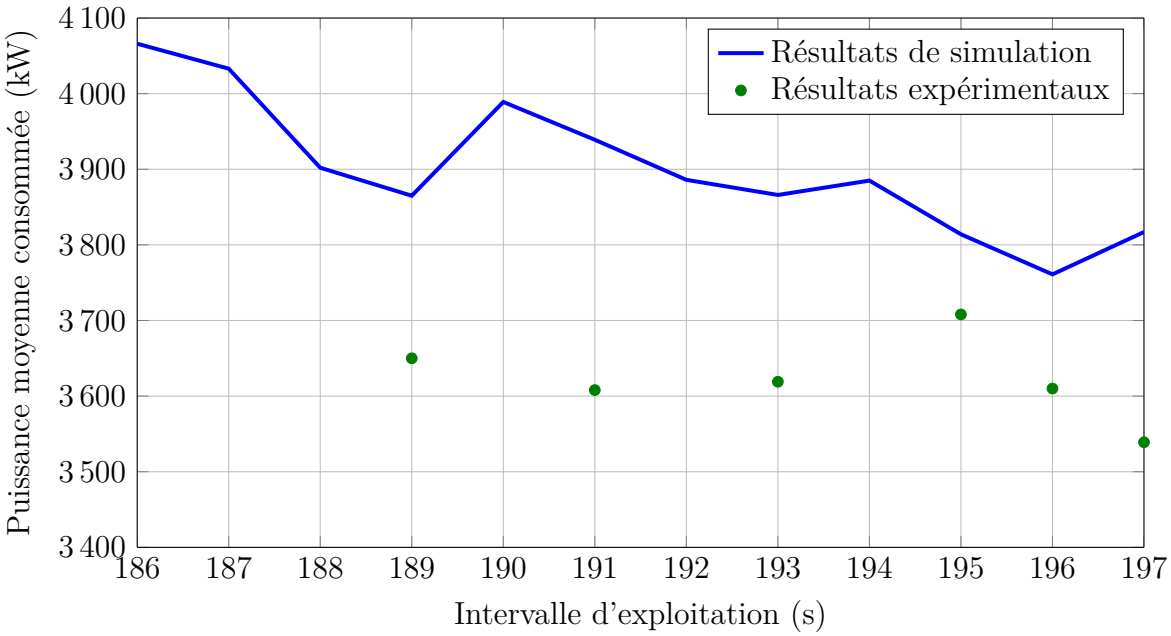


FIGURE 3.9 – Comparaison des résultats de simulation avec des mesures sur le site de Turin pour un carrousel composé de 16 trains

3.4 Optimisation des temps d'arrêt en station

3.4.1 Principe de la modulation des temps d'arrêt en station

Dans la section précédente, une étude de l'influence de l'intervalle d'exploitation sur la consommation électrique de la ligne a été menée. Pour aller plus loin, il convient également d'étudier l'impact de la modulation des temps d'arrêt en station sur cette consommation.

Afin de faciliter la visualisation du problème à solutionner, une représentation de ce dernier est donnée par la figure 3.10. Sur cette figure, les profils théoriques de puissance électrique consommée par cinq trains en exploitation sur la ligne de Turin ont été représentés.

L'objectif de la modulation des temps d'arrêt en station est de modifier la longueur des phases d'arrêt en station (phases rouges) pour faire coïncider le maximum de phases d'accélération (phases bleues) avec des phases de freinage (phases vertes) afin de maximiser la réutilisation de l'énergie électrique générée lors du freinage. Les profils présentés en figure 3.10 sont des profils théoriques puisque les phases de freinage sont constituées de paraboles indiquant un renvoi total de l'énergie issue du freinage ; dans un cas réel d'exploitation, l'allure de la puissance électrique renvoyée est généralement beaucoup plus irrégulière puisque les conditions de réceptivité totale de la ligne sont rarement rencontrées.

3.4.2 Estimation de l'espace des solutions

Dans le cadre de l'optimisation des temps d'arrêt en station, une solution est définie comme la combinaison des temps d'arrêt effectués par les trains en exploitation. Le nombre de solution possibles est déterminé par (3.12) où A_{mod} représente l'amplitude de la modulation et n_{stop} le nombre d'arrêt en station à optimiser.

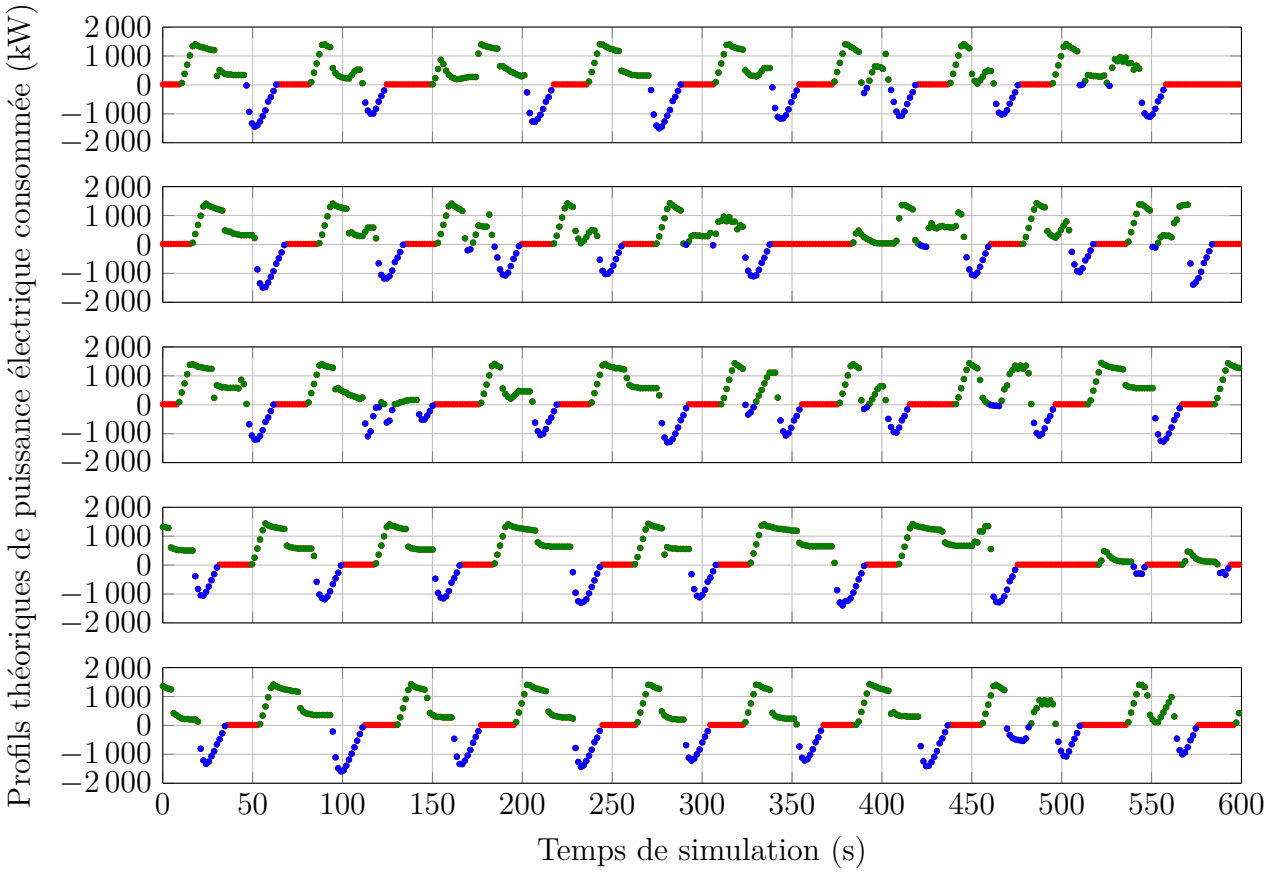


FIGURE 3.10 – Consommation énergétique théorique de cinq trains en exploitation.

$$N_{sol} = {A_{mod}}^{n_{stop}} \tag{3.12}$$

En appliquant cette définition au cas d'étude présenté à la figure 3.10, le nombre de solutions possibles est de $5^{42} \approx 2,3.10^{29}$ (en considérant une amplitude de modulation de ± 2s). De fait, le problème d'optimisation des temps d'arrêt en station possède une grande complexité liée à un espace des solutions de grande taille qui croît exponentiellement avec l'horizon des événements considérés.

Le problème d'optimisation des tables horaires a d'ailleurs été qualifié de NP-complet dans de nombreuses études comme [23], [77], [78] ou [79]. [75] en fait même une démonstration mathématique dans ses travaux de thèse.

Ainsi, dans ce type de problème, les méthodes de résolution exactes classiques sont applicables mais nécessitent d'importantes ressources de calculs. Des heuristiques sont alors généralement développées pour trouver des solutions approchées de l'optimum en un temps raisonnable [23], [80].

3.4.3 Détermination de la méthode d'optimisation

Afin de déterminer le moyen le plus approprié pour résoudre le problème d'optimisation des temps d'arrêt en station, il convient d'inspecter l'ensemble des méthodes d'optimisation (figure 3.11). Cette figure est tirée des travaux de [81] et [82], dans lesquels sont passées en revue un certain nombre de méthodes d'optimisation pour déterminer celles qui sont les plus appropriées pour les problèmes que les auteurs considèrent.

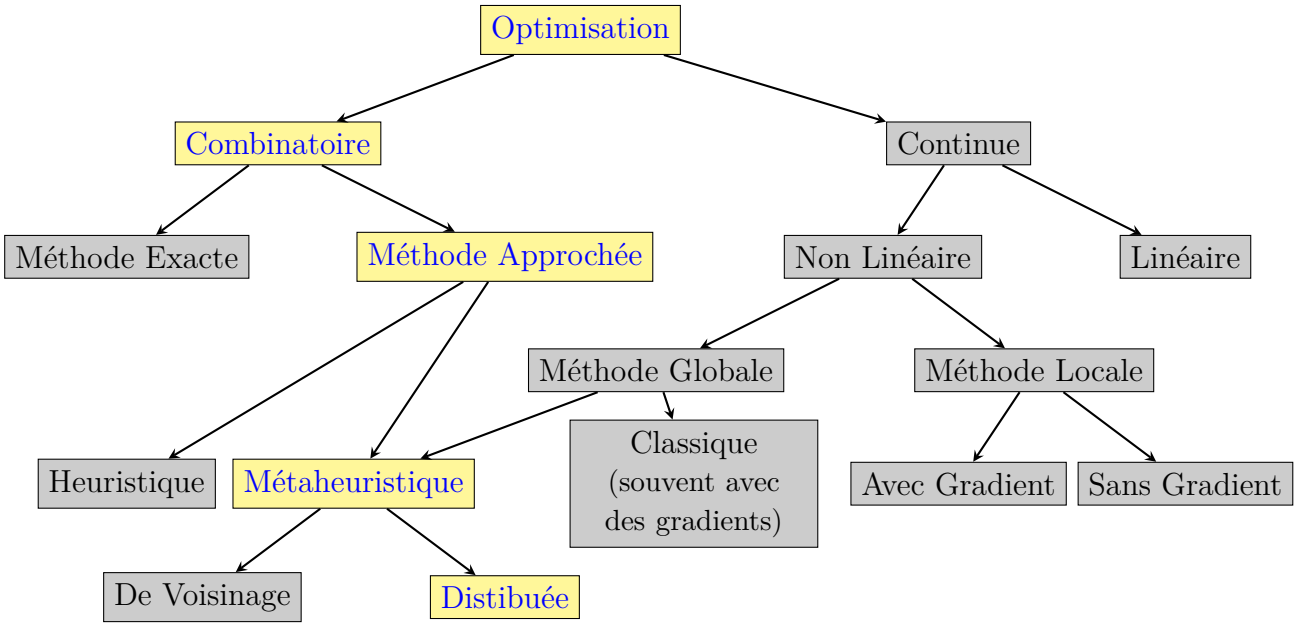


FIGURE 3.11 – Apercu des méthodes d'optimisation.

D'après la figure 3.11, le choix de la méthode d'optimisation se fait selon plusieurs critères : le type des variables du problème, la connaissance d'un modèle mathématique du problème, la connaissance de l'évolution de la fonction objectif sur l'espace des solutions, ou encore la possibilité d'utiliser une population de solutions pour effectuer la résolution.

La modulation des temps d'arrêt en station fait appel à des variables discrètes : les temps d'arrêt appartiennent à l'ensemble des entiers naturels $\mathbb{N}$. En présence d'un problème d'optimisation combinatoire difficile, une méthode approchée est alors privilégiée. Celle-ci peut faire soit appel à une heuristique spécialisée (entièrement dédiée au problème considéré), soit à une métaheuristique [1]. En outre, comme le rappelle [83], pour la majorité des problèmes d'optimisation combinatoire, aucun algorithme universel de résolution en temps polynomial n'est connu actuellement, ce qui renforce le choix d'utiliser une méthode approchée. Parmi les métaheuristiques, les métaheuristiques dites *de voisinage* font évoluer une seule solution à la fois tandis que les métaheuristiques dites *distribuées* font progresser en parallèle une population de solutions.

Selon le théorème du *no free lunch* [84], aucune méthode ne permet de résoudre tous les types de problèmes d'optimisation existants : si une méthode est meilleure qu'une autre pour une classe de problème alors elle sera moins bonne pour une autre classe. Il est alors important de se référer aux méthodes d'optimisation qui ont fait leur preuve dans la littérature pour résoudre une classe de problème similaire au problème courant.

Ainsi, au vu de la taille de l'espace des solutions, une méthode de voisinage de type recherche tabou ou recuit simulé ne permettrait pas d'effectuer suffisamment d'obser-

1. Il est intéressant de noter qu'un certain nombre de travaux traitant de la planification ferroviaire résolve ce problème par une programmation linéaire ou non linéaire à l'aide de variables continues. Cependant, le nombre important de minima locaux impose le recours à des méthodes d'optimisation globales et aux métaheuristiques dans le cas où les propriétés mathématiques de la fonction objectif ne sont pas totalement connues.

vations par rapport au nombre total de possibilités pour avoir une bonne estimation de la (ou les) solution(s) optimale(s) [85].

En outre, l'utilisation d'une métaheuristique distribuée se justifie également par le fait que les solutions manipulées sont de grandes dimensions et que l'utilisation d'une population de solutions permet de mieux analyser l'impact de chacune des dimensions de la solution sur la valeur finale de la fonction objectif.

Pour ces raisons, une résolution par métaheuristique distribuée est envisagée pour solutionner le problème d'optimisation des temps d'arrêt en station.

3.4.4 Optimisation par métaheuristique

Certaines heuristiques sont développées spécialement pour résoudre des problèmes spécifiques, d'autres sont quant à elles déployées pour solutionner plusieurs types de problèmes, ces algorithmes sont alors appelés métaheuristiques.

Dans la plupart des cas, une métaheuristique est définie comme un algorithme stochastique itératif. Les algorithmes stochastiques sont des méthodes utilisées dans des espaces de grandes dimensions pour simuler des lois de probabilité complexes. Le processus itératif permet de pouvoir évaluer l'impact des changements effectués à chaque itération afin de réorienter la recherche dans la direction qui permettrait a priori de converger vers l'optimum global avec la plus grande probabilité.

Un grand nombre de métaheuristiques sont inspirées de phénomènes naturels comme la biologie pour les algorithmes génétiques, l'éthologie pour les essaims particulaires (OEP) et les colonies de fourmis (ACO), ou encore la physique pour la méthode du recuit simulé [86], [87].

Ces méthodes sont souvent non-déterministes, c'est à dire, que la convergence de l'algorithme vers une solution optimale n'est pas garantie. Cependant, cette contrainte est compensée par leurs capacités à explorer l'espace de recherche efficacement en quête d'optima locaux du problème.

Dans la suite des travaux, un intérêt tout particulier est porté aux métaheuristiques distribuées ou à population de solution pour leur capacité à faire évoluer un ensemble d'individu simultanément, afin que chaque solution bénéficie de l'expérience acquise par le groupe lors de l'exploration de l'espace des solutions.

Principalement, deux catégories de métaheuristiques à population coexistent : les algorithmes évolutionnaires et les algorithmes d'intelligence en essaim [88].

3.4.5 Algorithmes évolutionnaires

En 1859, Charles Darwin présente ses théories sur l'évolution des espèces qui se sont adaptées progressivement à leur milieu naturel sous l'influence de contraintes extérieures [89], [90]. Le principe fondamental de ces théories est que les individus les mieux adaptés à leur environnement survivent, se reproduisent et leur patrimoine génétique est transmis à leur descendance. Ainsi, bien qu'initialement controversées, ces théories se sont peu à peu imposées, jusqu'à ce que l'analogie entre la sélection naturelle et l'optimisation soit mise en évidence à partir de la deuxième moitié du XXème siècle.

En effet, dans la conception Darwinienne de la sélection naturelle, l'adaptation des individus permet de faire face aux contraintes du milieu extérieur, tandis que dans une optimisation classique, les individus (ou solutions) sont modifiés itérativement, jusqu'à

ce qu'ils vérifient les contraintes du système ou permettent d'atteindre l'objectif initial.

Dès les années 1950, avec l'avènement des super-calculateurs, plusieurs travaux ont repris ce principe pour étudier l'évolution de systèmes soumis à des conditions variables et pour modéliser des processus évolutionnaires afin d'en déduire des outils d'optimisation pour les problèmes d'ingénierie courants.

Rechenberg propose ainsi de modéliser les paramètres influant sur l'évolution d'un système grâce à des *stratégies d'évolution* afin de résoudre des problèmes à variables continues [91] [2], Fogel introduit quant à lui la *programmation évolutionnaire* pour concevoir des automates à états finis [92].

Ces méthodes sont assez similaires à la version actuelle des *algorithmes génétiques* (AG), en effet, les opérateurs de sélection, la notion de génération et la notion de filiation étaient déjà présents. Cependant, la vraie innovation introduite par Holland en 1975 est d'utiliser une large population d'individus, ainsi que sa volonté de traduire les mécanismes d'adaptation naturelle en langage informatique [93].

Par la suite l'*évolution différentielle* [94] et la *programmation génétique* [95], apportent des améliorations à la méthode des algorithmes génétiques pour faciliter leur implémentation et leur applicabilité à un plus grand spectre de problèmes. Tous ces travaux sont regroupés dans la famille des *algorithmes évolutionnaires*.

Dans les algorithmes évolutionnaires, les solutions possibles du problème sont appelées individus et sont codés de manière analogue à des chromosomes. Un chromosome étant une représentation d'une solution, il peut être par exemple codé en binaire, dans ce cas, il s'agit d'une succession de 0 et de 1 appelés "bits" ou "gènes".

L'optimisation s'effectue par une succession d'itérations appelées générations et consiste à faire évoluer une population d'individus grâce à des opérateurs de sélection naturelle pour générer une nouvelle génération d'individus. Les performances des individus sont alors évaluées par la valeur de leur fonction objectif ou *fitness* qui reflète la capacité des individus à voir leurs génotypes être propagés à la génération suivante.

Plus généralement, la fonction objectif caractérise le degré d'atteinte de l'objectif de l'optimisation.

3.4.5.1 Optimisation par algorithme génétique

Parmi tous les algorithmes évolutionnaires évoqués précédemment, les algorithmes génétiques sont la technique la plus couramment employée dans les travaux d'optimisation. Ainsi, [31], [34], [67] et [69] mettent en œuvre cette méthode pour mener l'optimisation de la consommation énergétique de lignes ferroviaires.

Dans sa version la plus simple, un algorithme génétique intègre trois types d'opérateurs évolutionnaires issus de la biologie : la sélection, la mutation et le croisement. L'algorithme 5 résume les étapes qui composent un algorithme génétique générique.

Un algorithme génétique classique débute avec une population de N_{pop} chromosomes dont les positions initiales sur l'hyper-espace des solutions sont tirées aléatoirement. A chaque itération, la fonction coût de chaque individu est évaluée, puis l'opérateur

2. Initialement, ces travaux avaient pour but d'optimiser la valeur de coefficients de surfaces aérodynamiques.

de sélection est appliqué pour choisir les chromosomes qui constitueront les individus parents.

Ensuite, l'opérateur de croisement génère les individus enfants à partir des individus parents. Enfin l'opérateur de mutation est alors appliqué à tous les chromosomes de la nouvelle population et la population finale est utilisée pour la prochaine itération de l'algorithme.

Algorithme 5 Algorithme d'optimisation par algorithme génétique

Entrée(s) Initialiser une population de N_{pop} individus

1: **Tant que** Le critère d'arrêt n'est pas vérifié **Faire**
2: Évaluer la valeur de fonction objectif de chacun des N_{pop} individus de la population.
3: Appliquer un opérateur de sélection pour choisir les N_{par} parents de la génération suivante
4: Sélectionner des couples de parents et appliquer un opérateur de croisement avec une probabilité P_{cross} pour générer des couples d'enfants
5: Appliquer un opérateur de mutation à chacun des enfants avec une probabilité P_{mut}
6: **Fin du Tant que**

Sortie(s) Population finale N_{pop} individus

3.4.5.2 Représentation des chromosomes

Pour les besoins de l'étude, les chromosomes des individus sont codés comme des chaînes d'entiers. Si on considère une ligne de métro à $N_{stations}$, chaque individu est composé de $2N_{stations}$ gènes correspondant aux temps d'arrêt dans les différentes stations. La figure 3.12 illustre la manière dont est codée chaque solution de la population, où s_n représente le temps d'arrêt effectué dans la station n. Chaque gène respecte également la contrainte (3.18).

s_1	s_2	$\cdots$	s_n	$\cdots$	s_{2n}

FIGURE 3.12 – Représentation chromosomique des individus.

3.4.5.3 Opérateurs de sélection

La sélection permet de déterminer quels chromosomes seront choisis pour se reproduire. Un ou plusieurs critères sont alors définis pour sélectionner les individus qui sont pressentis pour converger vers une solution optimale. Généralement, le critère de décision est la fitness de la solution.

Différents opérateurs de sélection peuvent être employés :

La sélection *uniforme* consiste à choisir les chromosomes parents selon un critère d'équiprobabilité : toutes les solutions possèdent la même probabilité d'être sélectionnées.

La sélection *élitiste* consiste à ne choisir que les individus présentant la meilleure valeur de fonction objectif. Ce processus est donc par définition déterministe,

puisque seuls les meilleurs individus sont sélectionnés, au détriment de la diversité génétique qui aurait pu permettre de produire de bonnes solutions dans les générations suivantes.

La sélection *tournoi* compare les fitness des individus deux à deux et choisit celui qui a le meilleur coût. Une probabilité de ne pas choisir le meilleur chromosome est introduite pour intégrer un aspect aléatoire à cette méthode de sélection.

La sélection *roulette* consiste à définir une probabilité de sélection pour chaque individu selon la valeur de sa fonction coût. L'algorithme utilisé pour définir cette sélection est présenté dans l'algorithme 6, où N_{pop} est le nombre d'individus composant la population et C_i le coût de la fonction objectif du i-ème individu.

Algorithme 6 Sélection "Roulette"

1: $S_1 = \sum_{i=1}^{N_pop} F_{eval}(C_i)$.
2: Choisir un nombre L tel que $L \in [0; S_1]$.
3: **Tant que** $S_2 < L$ **Faire**
4: $\quad i \leftarrow i + 1$
5: $\quad S_2 \leftarrow S_2 + F_{eval}(C_i)$
6: **Fin du Tant que**
7: Return i

La sélection élitiste est le seul opérateur déterministe, les autres sont qualifiées de stochastique car les mauvaises solutions possèdent une probabilité d'être sélectionnées pour former la population de parents.

Dans notre cas d'étude, une hybride de sélection élitiste et de sélection par tournoi a été mise en œuvre selon la règle (3.13), où α_1 et $\alpha_2 \in [0, 1]$ représentent les coefficients de pondération entre les deux méthodes choisies et les variables *elitiste* et *tournoi* sont respectivement les règles de sélection élitiste et par tournoi.

Cette hybridation permet d'une part de conserver le patrimoine génétique des meilleurs individus d'une génération à l'autre, et d'autre part d'introduire une notion de brassage génétique en rendant une partie de ce processus stochastique grâce à la sélection par tournoi.

$$Selection(N_{pop}) = \alpha_1 \cdot elitiste(N_{pop}) + \alpha_2 \cdot tournoi(N_{pop}) \tag{3.13}$$

3.4.5.4 Opérateurs de croisement

Le croisement est une opération qui permet à deux chromosomes parents de s'échanger une ou plusieurs séquences de gènes afin de créer deux chromosomes enfants. Dans la littérature, un grand nombre de travaux préconisent d'effectuer un croisement multi-points pour augmenter le processus de brassage génétique, où N_{points} est le nombre de points de croisement.

Il est à noter que plusieurs terminologies comme les termes *enjambement* et *recombinaison* sont également utilisés pour décrire le phénomène de croisement. En outre, une probabilité d'occurence de croisement P_{cross} est définie pour conserver le caractère stochastique de ce processus.

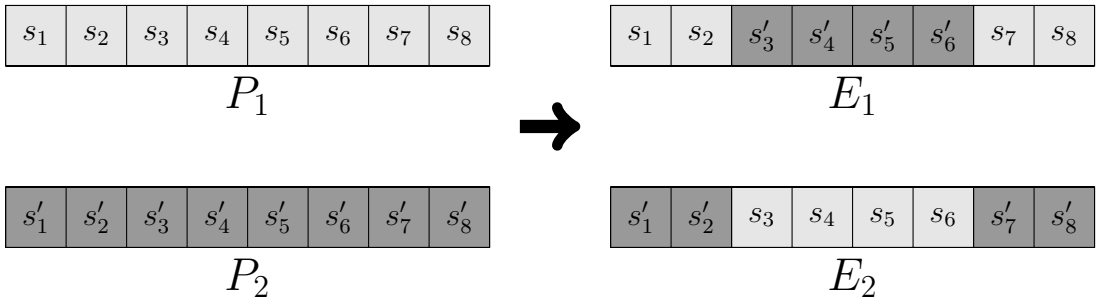


FIGURE 3.13 – Exemple d'utilisation de l'opérateur de croisement.

L'opérateur de croisement est illustré par la figure 3.13, avec 2 chromosomes parents P_1 et P_2 composés de 8 gènes qui donnent naissance à 2 chromosomes enfants E_1 et E_2. Dans cet exemple 2 points de croisements aux loci 2 et 6 sont utilisés.

3.4.5.5 Opérateurs de mutation

La mutation est une opération génétique où un ou plusieurs gènes du chromosome voient leurs valeurs être modifiées. Comme pour le croisement, une probabilité d'occurrence des mutations P_{mut} est introduite. Ce paramètre doit être défini avec attention pour ne pas transformer ce processus en une recherche aléatoire.

L'opérateur de mutation a pour objectif de diversifier la population pour explorer l'espace des solutions, tout en évitant que l'algorithme ne reste bloqué dans un extremum local. Cet opérateur assure donc une diversification supplémentaire du patrimoine génétique des individus.

Individu sélectionné — Individu muté

FIGURE 3.14 – Exemple d'utilisation de l'opérateur de mutation.

La figure 3.14 illustre l'opérateur de mutation sur un chromosome composé de 8 gènes, où les gènes situés aux loci 4 et 7 subissent une modification de leur valeur. Chaque gène de la chaîne chromosomique a une probabilité P_{mut} de voir sa valeur modifiée après application de l'opérateur de mutation.

3.4.5.6 Paramétrage de l'algorithme

Les valeurs des coefficients α_1, α_2, P_{mut}, P_{cross}, N_{pop}, N_{points}.... sont déterminées empiriquement par une série d'essais et d'erreurs qui ont pour but de déterminer le paramétrage permettant de maximiser la vitesse de convergence et le taux de convergence.

Dans cette étude, l'algorithme développé a pour objectif de garder une population homogène tout au long de l'optimisation de sorte à effectuer une recherche simultanée sur les différentes dimensions du problème pour qu'à l'issue de l'optimisation, une grande partie de l'espace des solutions ait été explorée.

En outre, pour augmenter la diversité génétique, une étape supplémentaire d'insertion de nouveaux individus est ajoutée après l'étape de mutation.

Cette étape d'insertion, permet d'explorer de nouvelles zones de l'espace des solutions tout en exploitant les solutions déjà obtenues par l'application des opérateurs évolutionnaires.

Le diagramme final de l'algorithme génétique utilisé dans ces travaux est résumé par la figure 3.15.

Le critère d'arrêt doit permettre de stopper l'algorithme soit quand la convergence de l'algorithme est atteinte, soit quand la meilleure solution explorée n'évolue plus sur plusieurs générations, ou quand le temps alloué à l'exploration de l'espace des solutions est dépassé.

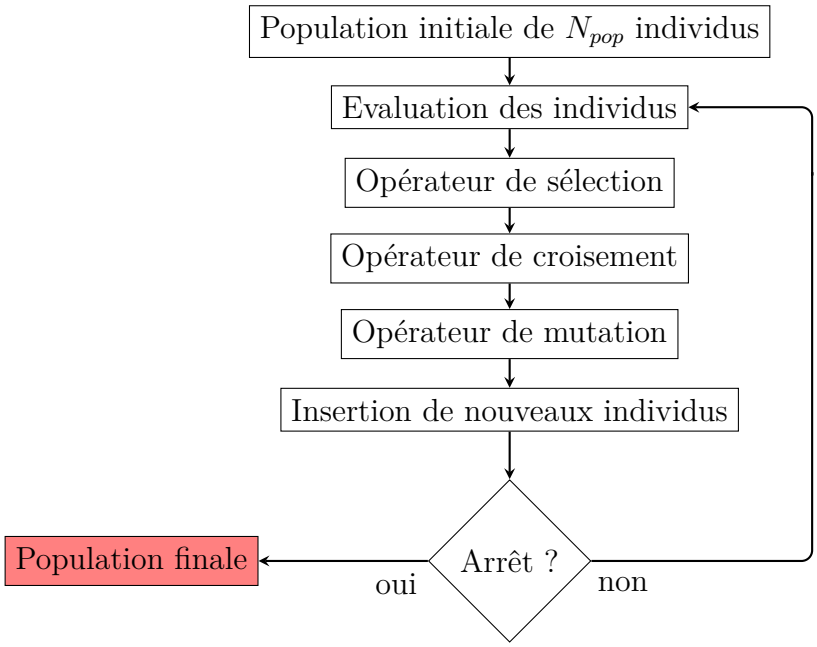


FIGURE 3.15 – Algorithme d'optimisation par algorithme génétique.

3.4.6 Algorithmes d'intelligence en essaim

Les algorithmes d'intelligence en essaim sont issus de l'étude du comportement collectif de certaines espèces comme les oiseaux, les poissons, les fourmis ou encore les abeilles. Par exemple, les fourmis laissent des traces de phéromones pour indiquer à leurs congénères le chemin vers les points d'eau et de nourriture, les abeilles s'agglutinent entre elles durant l'hiver pour se protéger du froid et se répartissent les tâches pour organiser la vie de la ruche, les oiseaux adoptent une formation en V pour minimiser les pertes aérodynamiques lors des longues migrations et les poissons adoptent une formation en banc serré notamment pour se protéger des prédateurs. La figure 3.16 présente quelques exemples d'intelligence en essaim pour différentes espèces animales.

Les algorithmes d'intelligence en essaim se caractérisent par l'utilisation d'une population d'agents. Ici, la notion d'agent se différencie de celle de solution, puisqu'ici chaque individu de la population a la possibilité de communiquer avec les autres membres de la population, et dispose de son propre système de décision.

L'intelligence de l'essaim est alors générée par des règles simples qui régissent les interactions entre les agents et leur environnement et aboutit à l'émergence d'un compor-

(a) Formation en V d'un groupe d'oiseaux.

(b) Banc de poissons.

(c) Pont flottant de fourmis.

FIGURE 3.16 – Exemple d'intelligence en essaim dans le règne animal.

tement pour l'essaim entier [85]. Un tel système est *auto organisé*, puisque le comportement de l'essaim n'est pas déterministe et dépendra des règles décisionnelles imposées aux agents.

Les algorithmes les plus populaires de cette famille de métaheuristique sont les méthodes d'optimisation par essaim particulaire (OEP), les colonies de fourmis et les colonies d'abeilles artificielles [96].

Parmi les méthodes citées précédemment, nous avons fait le choix de mettre en œuvre l'optimisation par essaim particulaire ou *particle swarm optimization*. Cette technique est régulièrement utilisée dans la littérature scientifique pour solutionner des problèmes d'optimisation dans une vaste gamme de domaines d'application.

3.4.6.1 Optimisation par essaims particulaires

Initialement, les travaux de Reynolds [97] et Heppner [98] ont permis de mettre en évidence que les animaux au sein d'un groupe en mouvement suivent le mouvement global du groupe grâce aux déplacements de leurs voisins tout en conservant un espacement optimal entre les individus.

Dans la nature, il est possible de remarquer que la dynamique de déplacement d'un groupe d'animaux peut être très complexe alors que chaque individu pris individuellement n'a qu'une connaissance limitée de sa position dans l'essaim (figure 3.16).

Par la suite, en 1995, Kennedy et Eberhart ont repris ces observations pour en déduire une métaheuristique de recherche : l'optimisation par essaims particulaires (OEP) [99]. Ainsi, l'OEP exploite ce concept de déplacement coordonné en intégrant une dimension sociale et une mémoire de groupe aux particules composant l'essaim afin de mener une recherche efficace. La stratégie globale de l'essaim s'adapte alors selon les expériences vécues par chaque particule de l'essaim pour atteindre l'optimum global de l'espace des solutions.

La popularité de cet algorithme s'explique, entre autres choses, par sa relative simplicité d'implémentation, le faible nombre de paramètres de réglages pour paramétrer le déroulement de la recherche, sa capacité à explorer efficacement un espace à $n-$dimensions, ou encore l'utilisation de règles de recherche simple qui n'utilisent pas de gradient.

Dans la pratique, les particules de l'essaim sont considérées comme des solutions possibles au problème. Une particule est définie par une position X_i^n sur l'espace des solutions et une vitesse de déplacement V_i^n [3], elle possède également une mémoire qui stocke la meilleure position visitée $P_{i,best}$ et la valeur de la fonction objectif en ce point. Les particules ont également la capacité de se communiquer entre elles la position de la meilleure solution globale connue de l'essaim G_{best}.

3.4.6.2 Règles de déplacement

La figure 3.17 représente le déplacement d'une particule dans un espace à 2 dimensions. Le déplacement est conditionné par trois composantes :

— Une composante inertielle qui entraîne la particule dans la direction de recherche courante héritée de l'itération précédente (traits bleus).

— Une composante cognitive qui pousse la particule à se diriger vers la meilleure solution qu'elle a visitée $P_{i,best}$ lors des itérations précédentes (traits rouges).

— Une composante sociale qui amène la particule à se diriger vers la meilleure solution G_{best} visitée par les autres particules de son voisinage (traits verts).

L'équation du mouvement est donnée par le système (4.30). ωV_i^n représente l'inertie de la particule, $r_1\beta_1(P_{i,best} - X_i^n)$ est la mémoire cognitive de la particule, tandis que le dernier terme $r_2\beta_2(G_{best} - X_i^n)$ exprime la mémoire sociale de l'essaim. Le coefficient d'inertie ω sert à contrôler l'importance de la direction de recherche courante sur le déplacement futur et pour régler la capacité d'exploration de l'essaim tandis que β_1 et β_2, respectivement le coefficient cognitif et le coefficient social, permettent de régler la capacité d'exploitation. r_1 et r_2 sont quant à eux des réels tirés uniformément dans l'intervalle $[0, 1]$. L'ensemble de ces coefficients a donc pour objectif d'orienter la recherche de l'optimum.

$$\begin{cases} V_i^{n+1} = \omega V_i^n + r_1\beta_1(P_{i,best} - X_i^n) + r_2\beta_2(G_{best} - X_i^n) & \text{(3.14a)} \\ X_i^{n+1} = X_i^n + V_i^{n+1} & \text{(3.14b)} \end{cases}$$

3. Le terme vitesse est un abus de langage puisque ce vecteur V n'est pas homogène à une vitesse, il s'agit plutôt d'une direction et d'une amplitude de déplacement.
L'utilisation du terme vitesse permet de conserver l'analogie avec le monde animal.

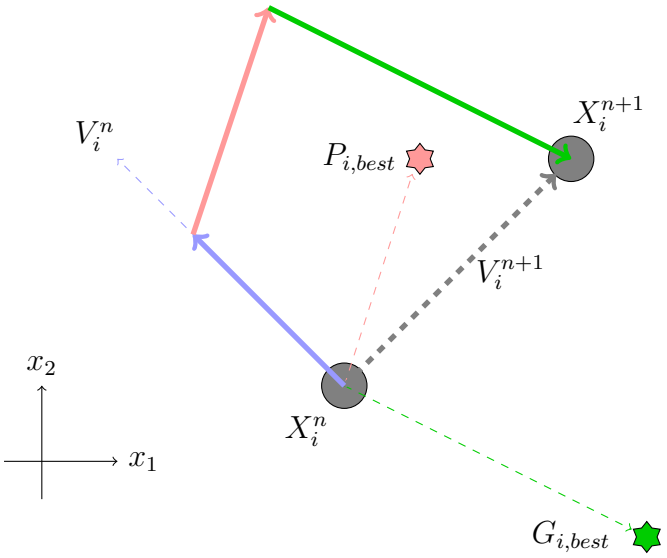


FIGURE 3.17 – Représentation du déplacement d'une particule dans un espace à 2 dimensions.

3.4.6.3 Codage des solutions

Chaque solution de l'essaim est codée de manière analogue à un chromosome, par une liste des temps d'arrêt qui seront effectués successivement par les trains en exploitation. Une particule est alors un vecteur d'entiers appartenant à un espace $n-$dimensions et bornés sur chaque dimension du problème. Ces bornes permettent, lors du processus itératif, de ne pas explorer les solutions qui violent les conditions d'exploitations définies en 3.2.3. L'objectif étant que chacune des positions visitées représentent un scénario de stationnement plausible.

3.4.6.4 Notion de voisinage

Au sein de l'essaim, les particules communiquent entre elles via un réseau social appelé voisinage. Il ne s'agit pas ici de voisinage géométrique mais plutôt d'un voisinage topologique. Un voisinage spatial nécessiterait de recalculer à chaque itération le nouveau voisinage géométrique des particules ce qui alourdirait la procédure de recherche. D'après [87], le choix de la topologie influence grandement les propriétés de convergence de l'OEP.

Principalement trois types de topologie de voisinage existent dans la littérature : le voisinage en rayon où toutes les particules ne communiquent qu'avec une particule centrale ; le voisinage en anneau où chaque particule n'est relié qu'avec un nombre limité d'autres particules, les particules tendent alors à se déplacer vers la meilleure particule de son voisinage ; le voisinage en étoile où toutes les particules sont capables de communiquer entre elles [86].

Ici, le choix a été fait d'une communication totale entre tous les individus de l'essaim. Ce choix est notamment motivé par le fait que dans notre cas, l'espace des solutions a une dimension assez élevée $dim_{Esol} \in [10^1; 10^3]$, de sorte qu'il y a peu de probabilités que les particules stagnent dans un optimum local. Cette version de l'algorithme d'OEP où tous les individus de l'essaim communiquent entre eux est appelée *version globale*, puisque toutes les particules ont connaissance d'un même optimal

global.

3.4.6.5 Limitation de la vitesse de déplacement

Lors du processus itératif, il peut arriver que la vitesse d'une particule devienne excessive ce qui pourrait l'amener à passer à côté d'un optimum et à sortir de l'espace des solutions admissibles.

Une solution évoquée par [87], considère de confiner l'espace de recherche en s'assurant soit que l'évaluation de la fonction objectif n'est effectuée que pour les particules se trouvant à l'intérieur de l'espace de recherche, soit que les particules restent dans l'espace de recherche en les arrêtant à la frontière, soit en utilisant un opérateur de rebond pour qu'une fois arrivées à la frontière de l'espace, les particules reviennent dans l'espace des solutions admissibles.

Dans le système (4.30), les termes ω, β_1 et β_2 permettent d'effectuer un compromis entre exploration et exploitation et d'après [100] et [101], l'établissement d'une relation de dépendance entre ces paramètres permettrait d'améliorer les propriétés de convergence de la méthode.

Les auteurs ont donc mis en place un *coefficient de constriction* Ψ. Ce coefficient présente l'avantage de se passer de la définition d'une vitesse maximale de déplacement tout en contrôlant l'amplitude du déplacement des particules. L'équation de mise à jour de la vitesse devient alors (3.15).

Cependant, nos divers essais pour mettre en oeuvre cette variante ne se sont pas avérés probants puisque le taux de convergence était dégradé en utilisant cette modification.

L'une des raisons les plus probables, de cette discordance par rapport aux travaux [100] et [101], serait un trop grand nombre de dimensions dans le problème d'optimisation considéré.

$$V_i^{n+1} = \Psi V_i^n + r_1\beta_1(P_{i,best} - X_i^n) + r_2\beta_2(G_{i,best} - X_i^n) \tag{3.15}$$

Pour ces raisons, un dimensionnement approprié du coefficient d'inertie ainsi que des paramètres cognitif et social est privilégié pour limiter et orienter le déplacement des particules.

Ainsi, la valeur du coefficient d'inertie détermine si les particules suivent une politique d'exploration globale ($\omega > 1$) ou au contraire une exploration locale ($\omega < 1$) en donnant plus ou moins d'amplitude au déplacement des particules.

Deux approches ont été testées : une approche utilisant un coefficient d'inertie fixe et une autre utilisant une règle de décroissance linéaire de ce coefficient (3.16).

Cette dernière approche a notamment était présentée dans les travaux de [87] et [102]. Dans (3.16), ω_{min} et ω_{max} représentent les bornes de variation du coefficient d'inertie, $n_{iteration}$ est l'indice de l'itération courante et $max_{iteration}$ est le nombre maximal d'itérations prévues pour la résolution du problème.

$$\omega = \omega_{min} + \omega_{max}\frac{n_{iteration}}{max_{iteration}} \tag{3.16}$$

L'utilisation d'un coefficient d'inertie fixe nécessite d'effectuer un grand nombre d'essais pour affiner la valeur de ce paramètre, tandis que l'emploi d'un coefficient

variable requiert de définir un nombre maximal d'itérations, ce qui peut s'avérer délicat en cas de convergence lente.

Néanmoins, en fixant le nombre d'itérations à une valeur élevée et en ajoutant un autre critère d'arrêt pour stopper l'exploration de l'espace des solutions, le coefficient d'inertie suit une décroissance progressive qui permet au fur et à mesure des itérations de passer d'une exploration globale à une exploration locale [4].

Dans notre étude, le critère d'arrêt supplémentaire vise à stopper le processus d'optimisation lorsque la meilleure solution connue par l'essaim n'évolue plus sur un certains nombre d'itérations, afin de ne pas prolonger le processus de recherche plus qu'il n'est nécessaire.

3.4.6.6 Implémentation de l'algorithme

L'algorithme 7 résume les étapes d'un algorithme classique d'optimisation par essaims particulaires, où f représente la fonction coût de l'algorithme.

Algorithme 7 Algorithme d'optimisation par essaims particulaires

1: Initialiser une population de N_{pop} individus, définis par des vitesses aléatoires et des positions prises dans l'espace des solutions.
2: Evaluer les valeurs de fonction objectif aux positions des N_{pop} individus.
3: Définir les optima initiaux $P_{i,best}$ et G_{best}.
4: **Tant que** Le critère d'arrêt n'est pas vérifié **Faire**
5: **Pour** i = 1 à N_{pop} **Faire**
6: Mettre à jour la vitesse de la particule suivant l'équation (3.14a)
7: Mettre à jour la position de la particule suivant l'équation (3.14b)
8: Evaluer la valeur de la fonction objectif à la position de la particule $f(X_i(n))$
9: **Si** $f(X_i(n)) < f(P_{i,best})$ **Alors**
10: $P_{i,best} \leftarrow X_i(n)$
11: **Fin du Si**
12: **Si** $f(X_i(n)) < f(G_{best})$ **Alors**
13: $G_{best} \leftarrow X_i(n)$
14: **Fin du Si**
15: **Fin du Pour**
16: **Fin du Tant que**

L'algorithme 7 est volontairement simpliste car il existe dans la littérature un très grand nombre de variantes qui influent sur un ou plusieurs paramètres et démontrent une amélioration des propriétés convergentes de leurs algorithmes.

De fait, le choix a été fait de développer une méthode généraliste qui a montré des propriétés de convergence satisfaisante dans un grand nombre de classes de problèmes, à condition que les paramètres de l'algorithme soient correctement dimensionnés [86].

4. L'exploration locale s'apparente alors dans ce cas de figure à une exploitation de la recherche effectuée durant les premières itérations.

3.4.7 Hybridation des méthodes d'optimisation

3.4.7.1 L'hybridation dans la littérature

La méthode d'optimisation par essaim particulaire présente l'avantage de posséder une mémoire collective partagée par l'ensemble des particules de l'essaim, cependant, cette mémoire collective peut entraîner des problèmes de convergence prématurée vers un optimum local lorsque le paramétrage du déplacement des particules ne leur permet pas d'explorer suffisamment l'espace des solutions [87].

A l'inverse, dans la méthode d'optimisation par algorithme génétique, les individus n'ont pas de mémoire des meilleures positions visitées, mais les opérateurs évolutionnaires permettent une exploration assez approfondie de l'espace des solutions.

Ainsi, si les probabilités d'occurrence de ces opérateurs ne sont pas maitrisées, il en résulte une exploration aléatoire de l'espace de recherche sans garantie de convergence vers l'optimum global du problème.

On peut alors estimer grossièrement que l'AG est une méthode qui explore l'espace des solutions, tandis que l'OEP exploite les résultats de la recherche. Intuitivement, une hybridation entre ces deux méthodes permettrait de trouver une alternative au compromis exploration/exploitation. Dans ce cas de figure, l'hybridation permet de capitaliser sur les points forts des méthodes pour accélérer la convergence vers l'optimum global de l'espace de recherche.

L'hybridation de techniques d'optimisation consiste à répartir différentes tâches de recherche entre plusieurs méthodes [103].

L'hybridation de méthodes d'optimisation est régulièrement pratiquée dans la littérature, que ce soit avec des métaheuristiques ou des méthodes exactes. Dans cette section, un état de l'art des hybridations entre un algorithme génétique et un algorithme d'essaim particulaire a été effectué. L'idée sous-jacente de cette étude est qu'en hybridant deux méthodes qui ont fait leur preuve sur des problèmes d'optimisation combinatoire NP-difficiles, une métaheuristique de recherche encore plus efficace sera obtenue.

[104] propose d'hybrider un algorithme d'optimisation par essaim particulaire en lui intégrant des règles évolutionnaires issues d'un algorithme génétique (AG) selon trois variantes :

Hybridation parallèle : l'OEP et l'AG effectuent leurs itérations en parallèle. Lorsque la position de G_{best} n'évolue pas sur plusieurs itérations, une opération de croisement est réalisée entre la particule G_{best} et des individus issus de l'AG.

Hybridation série : l'optimisation par AG sert de point de départ à l'OEP. Une première optimisation est réalisée par algorithme génétique sur un nombre donné d'itérations puis, une optimisation par OEP est effectuée sur un même nombre d'itérations. La population finale issue de l'AG est utilisée comme population initiale de l'OEP.

Hybridation d'insertion : seul un algorithme d'OEP est employé pour réaliser l'optimisation. Lorsque la valeur de $P_{i,best}$ de la particule i n'évolue pas sur un certain nombre d'itérations, un opérateur de mutation est appliqué à la position de $P_{i,best}$.

Dans [104], l'auteur justifie le choix de l'utilisation de règles évolutionnaires dans l'OEP par le fait que le contrôle de la convergence et du déplacement des particules est

beaucoup plus important avec le coefficient d'inertie qu'avec les opérateurs de croisement et de mutation. En outre, chacune de ces variantes a été testé sur un *benchmark* de fonctions continues. Cependant, aucune variante n'a prouvé sa supériorité par rapport aux autres sur chaque problème du *benchmark* (conformémment au théorème du *No free lunch*).

Dans [105], l'auteur étudie plusieurs techniques d'hybridation entre différentes métaheuristiques à base de population. L'hybridation entre AG et OEP est réalisée par insertion en remplaçant l'étape de mutation des individus par les mécanismes de mise à jour de la PSO. Cette hybridation a pour objectif de réduire le temps de calcul de l'AG tout en améliorant la convergence de la méthode.

Une approche légèrement différente est explicitée dans [106], où un opérateur de mutation est ajouté à un OEP après l'étape de mise à jour de la position des particules pour augmenter l'exploration de l'espace de recherche.

Dans [107], une hybridation entre OEP et AG est également décrite. A chaque itération, les individus issus de l'optimisation par AG et OEP sont triés par valeur de fitness. Les meilleurs individus de la population sont utilisés par l'AG tandis que le déplacement des moins bons individus est régi par l'OEP. La routine de l'algorithme génétique est exécutée en premier de sorte que l'OEP puisse utiliser la meilleure position trouvée par l'AG pour la mise à jour des positions.

Il est enfin à noter que [87] propose de partitionner l'espace de recherche en assignant à chaque zone de l'espace une population dédiée. Bien que cette idée soit en théorie très judicieuse, en pratique, cela s'avère compliqué à mettre en œuvre quand le nombre de dimensions du problème augmente.

3.4.7.2 Choix d'hybridation retenu

A la lumière des travaux précédents, il a été décidé d'effectuer une hybridation en combinant les caractéristiques de l'hybridation parallèle et de l'hybridation série.

Le principe d'hybridation retenu est qu'à chaque itération, la population issue de l'optimisation par OEP a connaissance de la meilleure position visitée par la population de l'AG, et inversement, l'AG a la capacité d'utiliser les solutions déterminées par OEP pour former de nouveaux individus. De cette manière, chaque population bénéficie des propriétés de recherche et de convergence de l'autre méthode.

Cette hybridation a été privilégiée pour plusieurs raisons. D'une part, l'utilisation de deux populations générées par deux méthodes d'optimisation différentes permet de conserver les propriétés de convergence de chacune des méthodes à chaque itération, d'autre part, le partage des informations entre OEP et AG augmente la vitesse de convergence globale et la diversité des solutions explorées.

Enfin, cette hybridation permet à l'OEP d'exploiter les meilleures solutions explorées par l'AG et ainsi de réaliser un compromis entre exploration et exploitation.

L'algorithme d'optimisation hybride OEP-AG est récapitulé par le diagramme 3.18. Il est à noter qu'à la première itération de l'algorithme, la population AG est la même que la population OEP.

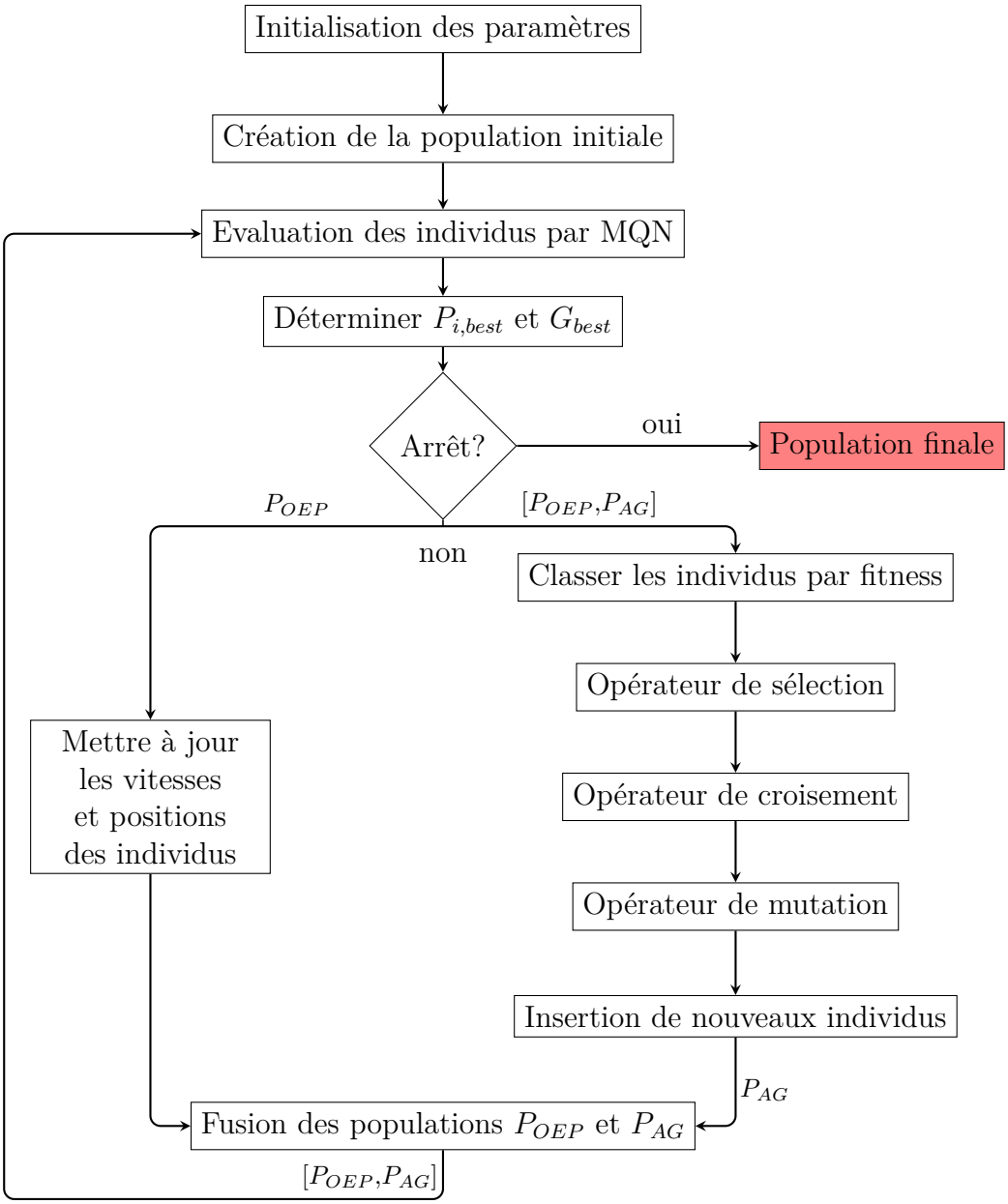


FIGURE 3.18 – Diagramme de l'algorithme hybride OEP-AG.

3.4.8 Comparaison des méthodes d'optimisation

Chacune des métaheuristiques présentées précédemment : OEP, AG et l'hybride OEP-AG a été implémentée en vue de solutionner le problème d'optimisation des temps d'arrêt en station. Ces méthodes ont ensuite été comparées pour déterminer laquelle est la plus efficace pour le type de problème rencontré.

L'indicateur de performance utilisé pour comparer ces techniques est le gain énergétique entre une table horaire utilisant des temps d'arrêt en station nominaux et le meilleur planning de stationnement trouvé par optimisation.

Lors du fonctionnement nominal d'un carrousel établi, sans aléas d'exploitation, les trains sont soumis à un intervalle et suivent une consigne de temps d'arrêt en station. Cette combinaison de temps d'arrêt en station effectuée par l'ensemble des trains du carrousel donne lieu à une consommation énergétique nominale.

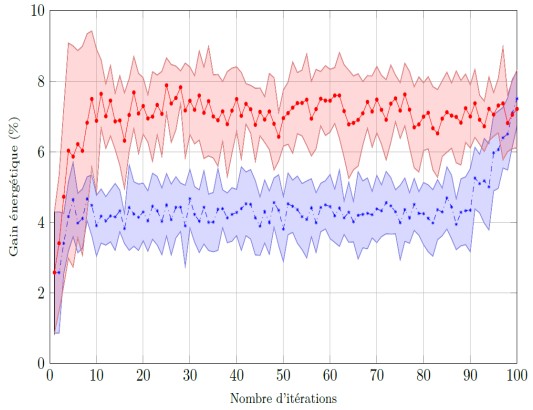
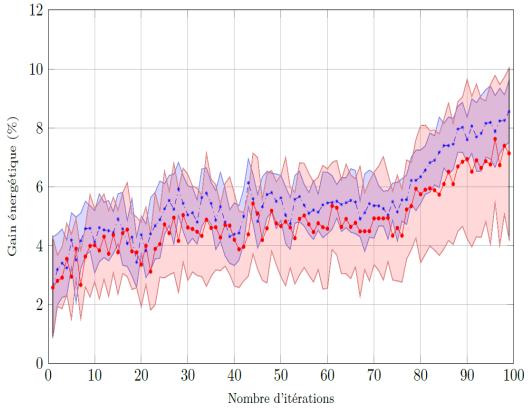
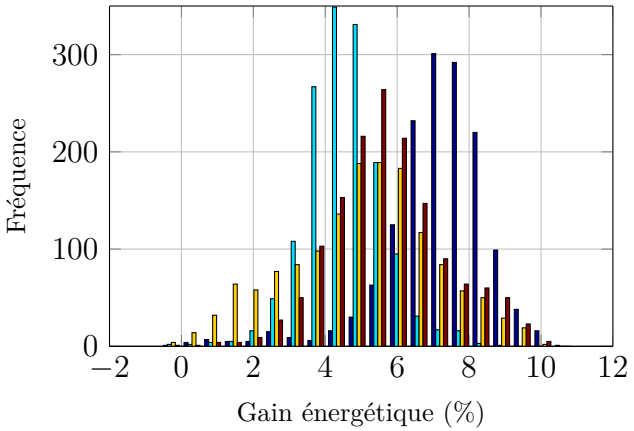
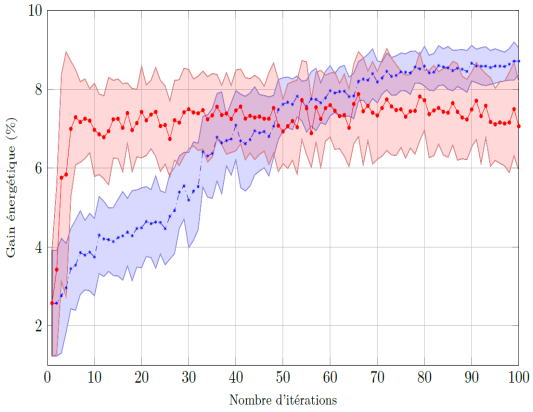
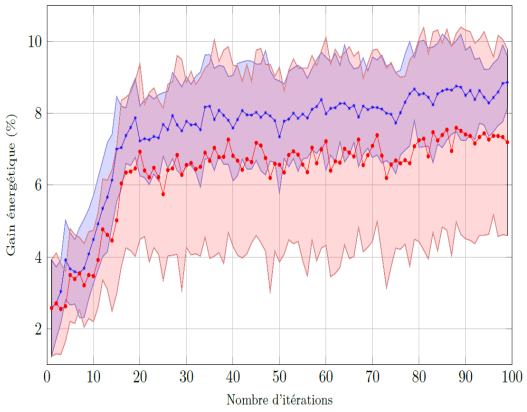
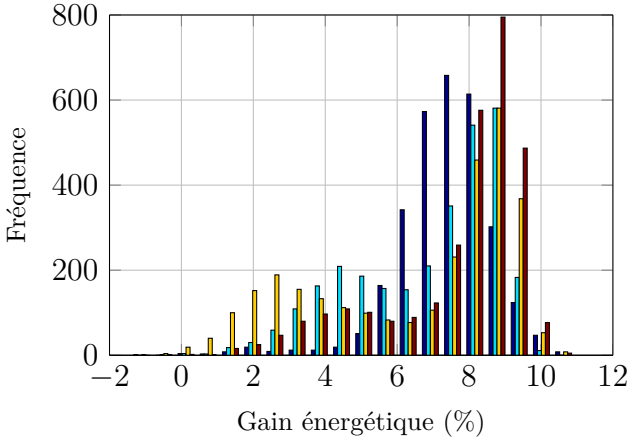
Les méthodes d'optimisation permettent de déterminer de nouvelles combinaisons

de temps d'arrêt qui auront une influence sur la consommation énergétique globale de la ligne de métro.

Le gain énergétique est donc défini par rapport à la consommation du carrousel sur un tour de boucle en considérant un fonctionnement nominal.

Le tableau 3.3 présente une comparaison des performances de convergence des populations de l'OEP et de l'AG et la densité de distribution des solutions renvoyées par chaque méthode. La colonne de gauche reprend le cas où chaque population évolue indépendamment l'une de l'autre ; tandis que la colonne du milieu correspond à l'évolution des populations dans le cadre de l'hybridation.

La colonne de droite analyse la distribution des solutions trouvées par chaque méthode afin d'analyser quelle méthode est la plus appropriée pour effectuer une recherche efficace.

N_{pop}	AG & OEP indépendants	Hybride OEP-AG	Densité de distribution
15			
30			

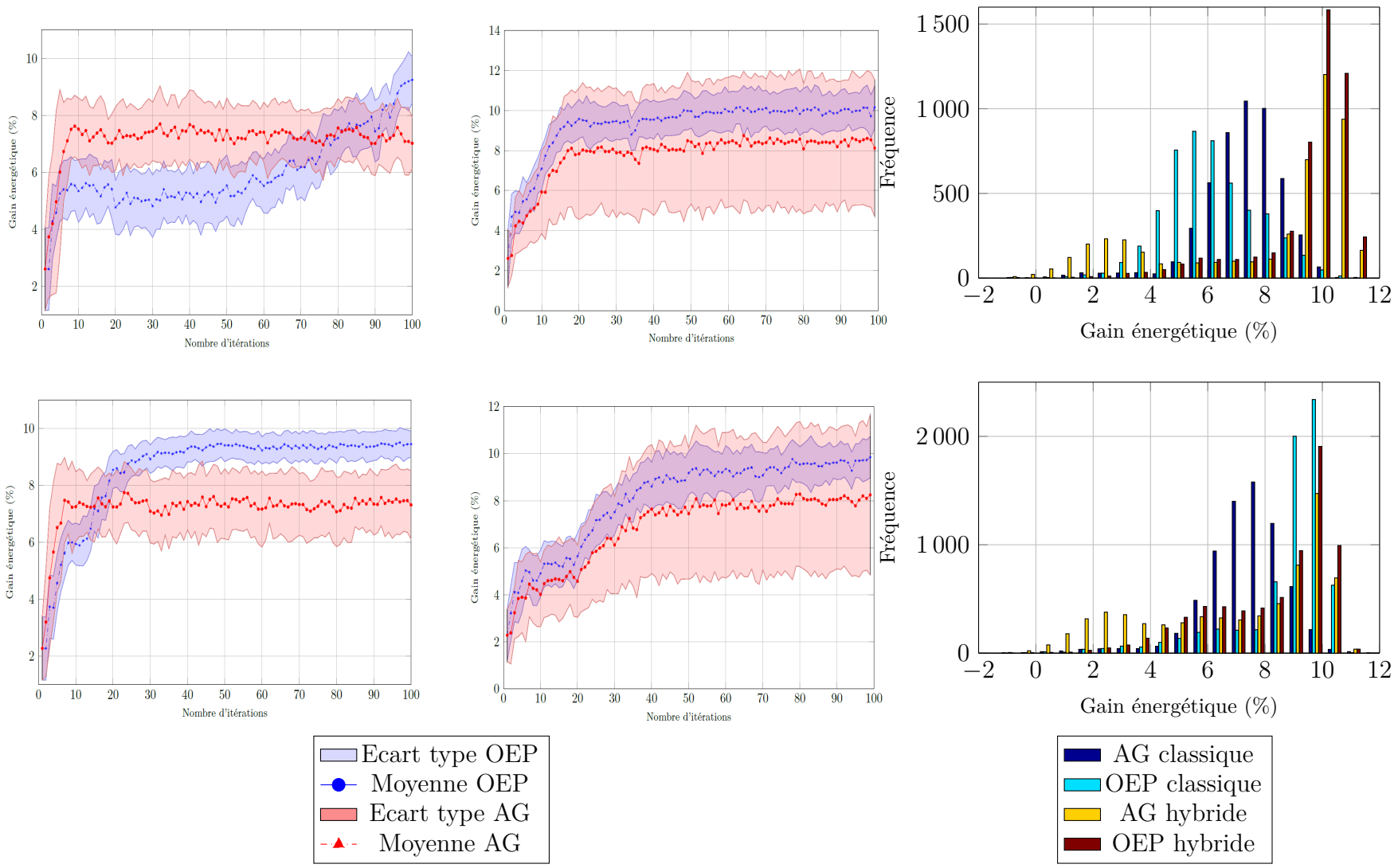


TABLEAU 3.3 – Comparaison des propriétés de convergence des méthodes d'optimisation en fonction de la taille de la population.

Le tableau 3.3 permet de relever quelques caractéristiques concernant le gain de performance obtenu en réalisant l'hybridation de la méthode AG-OEP :

— Pour un même nombre d'itérations et un même nombre d'individus dans la population, la méthode hybride permet de trouver des combinaisons de temps d'arrêt qui entraînent un plus fort gain énergétique.

— Dans sa version classique, les performances de l'AG semblent être indépendantes de la taille de la population à l'inverse de l'OEP.

— Lorsque les méthodes OEP et AG évoluent indépendamment, l'AG est plus performante que l'OEP durant les premières itérations. Cependant lorsque le nombre d'individus de la population augmente, l'OEP permet d'explorer de meilleures solutions. En outre, l'AG semble converger vers un optimum local dès les premières itérations, tandis que l'OEP voit une amélioration constante des solutions explorées.

— Lorsque les méthodes OEP et AG coopèrent en partageant leurs informations sur les solutions explorées, l'OEP domine l'AG dans tous les cas de figure simulés. En outre, l'écart type des solutions analysées par l'algorithme génétique est beaucoup plus étendu que celui des solutions explorées par essaim particulaire. L'hybridation assure un compromis entre exploration de l'espace et exploitation des résultats. L'AG joue alors le rôle de méthode exploratrice tandis que l'OEP exploite les solutions trouvées aux itérations précédentes pour converger vers un optimum.

D'après le tableau 3.3, une méthode OEP-AG hybride composée de 50 individus semble être la plus appropriée pour obtenir une distribution de solutions de bonne qualité tout en limitant le temps de calcul.

Le tableau 3.3 s'est intéressé à l'étude des performances d'amélioration de la fonction objectif, cependant, un autre aspect important à prendre en compte dans ces travaux est la performance en temps de calcul. Ainsi, le tableau 3.4 présente le temps de calcul moyen d'un individu sur une itération pour les 3 méthodes introduites précédemment, autrement dit le temps de calcul nécessaire à l'exploration d'une solution. Il est à noter que lors des différents essais, le temps de calcul de chacun des cas présentés dans le tableau 3.3 évolue linéairement en fonction du nombre d'individus composant la population.

	Méthode AG	Méthode OEP	Méthode hybride OEP-AG
Temps de calcul unitaire (s)	6.8613	6.7158	14.6305

TABLEAU 3.4 – Temps de calcul moyen d'une solution

D'après le tableau 3.4, une utilisation séquentielle des méthodes AG et OEP est plus rapide que la résolution du problème par l'hybride, avec un gain en temps de calcul d'environ 7.8%. Les temps de calcul correspondent à un PC embarquant un processeur Intel Xeon W3520 cadencé à 2.67GHz, 12 Go de RAM, une carte graphique Quadro FX 3800 avec 1Go de VRAM et le tout sous un environnement Windows 7.

Cette étude des performances des méthodes d'optimisation illustre donc le fait que l'amélioration des propriétés de convergence d'une méthode se fait par dégradation du temps de calcul/exploration.

3.4.9 Répartition des temps de calcul

D'après le tableau 3.4, le temps de calcul moyen d'une solution par la méthode hybride OEP-AG est d'environ 14.6s. Une étude plus approfondie de la répartition du temps de calcul nécessaire à l'exploration d'une solution montre qu'en moyenne 91% du temps de calcul est dédié à la résolution itérative des flux de puissance relatifs à la solution explorée.

Les 9% du temps de calcul restant concernent l'exécution des autres modules algorithmiques impliqués dans la boucle d'optimisation comme la construction du vecteur solution, la mise à jour des solutions générées par OEP, la mise à jour des solutions générées par AG,...

3.4.10 Remarques

Une démarche explorant à la fois l'intervalle d'exploitation et les temps d'arrêt en station aurait pu être adoptée, cependant, diverses expérimentations ont montré que cette approche était plus coûteuse en temps de calcul et fournissait exactement les mêmes résultats qu'une approche décorrélant chacune des étapes de l'optimisation des paramètres d'exploitation.

La taille de l'espace des solutions dans une démarche corrélant les deux paramètres d'exploitation serait donnée par (3.17a) alors qu'en utilisant une approche décorrélant ces deux paramètres, la taille de l'espace des solutions du problème global est donnée par (3.17b), où N_{inter} est le nombre d'intervalles possibles pour chaque carrousel tel que défini par (3.3).

$$N_{sol\,glob} = \begin{cases} {N_{inter}}^{{A_{mod}}^{n_{stop}}} & \text{(3.17a)} \\ {N_{inter}} \cdot {{A_{mod}}^{n_{stop}}} & \text{(3.17b)} \end{cases}$$

3.5 Optimisation d'une table horaire journalière

Dans le domaine ferroviaire, une table horaire journalière définit les horaires de début et de fin d'exploitation, les horaires de départs et d'arrivées de chaque train dans les différentes stations, les horaires d'insertion et de retrait des trains, les temps d'arrêt en station et les intervalles d'exploitation. Certaines de ces informations sont redondantes, mais permettent d'avoir une vision claire des conditions d'exploitation de la ligne.

Les tables horaires sont conçues à long terme pour chaque jour de la semaine et suivent la courbe d'affluence des passagers afin d'assurer la fluidité du transit des usagers.

Une table horaire définit également les durées et l'alternance d'exploitation entre phases établies et phases transitoires. Le terme *phase établie* désigne la période d'exploitation où l'ensemble des trains composant le carrousel sont espacés d'un même intervalle. Le passage d'un carrousel établi à un autre se fait par injection ou retrait de trains et est désigné par le terme *phase transitoire*.

Lors de la conception de la grille horaire, l'exploitant définit pour chaque phase établie, l'intervalle d'exploitation et les temps d'arrêt en station nominaux, cependant,

comme l'ont montré les travaux exposés en section 3.2.2, des gains substantiels peuvent être réalisés en modifiant la valeur de ces temps d'arrêt.

Ainsi, cette section a pour ambition d'utiliser la méthode hybride AG-OEP pour résoudre le problème d'optimisation des temps d'arrêt en station dans le cas d'une table horaire journalière, afin d'étudier les gains énergétiques réalisables à l'issue d'une journée d'exploitation type.

3.5.1 Phases transitoires : notion de train tenant l'horaire

3.5.1.1 Principe de l'injection/retrait

L'injection et le retrait automatique des trains aux terminus de la ligne ont pour objectif d'optimiser à chaque instant le nombre de rames en ligne théoriquement utile pour assurer le programme d'exploitation.

L'augmentation du nombre de rames nécessaires en ligne est obtenue en insérant dans le carrousel une nouvelle rame présente sur le parking d'injection du terminus, dans l'éventualité où aucune des rames présentes en ligne et arrivant à ce terminus ne peut assurer le prochain départ.

La diminution du nombre de rames nécessaire en ligne est, quant à elle, obtenue en retirant une rame lorsque le nombre de rames en ligne arrivant au terminus est supérieur au nombre de rames nécessaires pour assurer les prochains départs.

Selon les configurations rencontrées, un parking d'injection/retrait peut être présent sur un seul ou sur les deux terminus de la ligne de métro.

3.5.1.2 Notion de train tenant l'horaire

La notion de train qui tient l'horaire permet à la régulation de trafic de déterminer à quel moment doit intervenir l'injection/retrait d'un train sur la ligne en comparant la table horaire de départ théorique avec la table horaire d'arrivée théorique au terminus de départ. Trois cas de figures se présentent :

Exploitation en phase établie : Si un seul train en ligne est capable d'assurer le prochain départ avec un retard inférieur au seuil Δr_{inj}.

Condition d'injection : Si aucun train en ligne n'est en mesure d'assurer le prochain départ prévu avec un retard inférieur au seuil Δr_{inj}.

Condition de retrait : Si deux trains sont en mesure d'assurer le prochain départ avec un retard inférieur à un seuil Δr_{ret}

Les seuils Δr_{ret} et Δr_{inj} sont définis de manière à minimiser les mouvements d'injection/retrait en terminus. De plus, le seuil Δr_{inj} doit être supérieur à Δr_{ret} pour ne pas générer une condition d'injection temporaire qui sera suivi par l'occurrence d'une condition de retrait.

3.5.2 Méthodologie d'implémentation

Initialisation : Une fois que les intervalles d'exploitation nominaux ont été spécifiés pour les différentes phases établies composant la table horaire journalière, il convient d'initialiser le processus d'optimisation des temps d'arrêt en station. Pour cela, une population initiale de solution est définie.

Chaque solution est une liste des temps d'arrêt qui seront effectuées par les trains en station ordonnée selon les horaires nominaux d'arrivée en station des trains.

Comme illustré par l'équation (3.18), les temps d'arrêt composant cette population initiale sont pris aléatoirement dans une plage de variation définie en accord avec l'exploitant pour respecter au maximum les contraintes d'exploitation.

$$s_{i,nom} - \Delta s_{1i} \leq s_{i,j} \leq s_{i,nom} + \Delta s_{2i} \tag{3.18}$$

L'étape d'initialisation nécessite également de paramétrer correctement l'algorithme d'optimisation pour permettre une amélioration continue des solutions optimales à chaque itération, mais également de définir le nombre d'individus composant la population initiale.

Processus d'optimisation : L'étape initiale de la méthode d'optimisation consiste à évaluer la population initiale pour déterminer l'impact de chaque solution sur la consommation énergétique de la ligne.

Dans le cas d'une phase établie, cette évaluation s'effectue en simulant un carrousel établi effectuant un tour de boucle à vitesse nominale et effectuant les arrêts spécifiés par la solution, puis en déterminant les flux de puissance instantanée qui se produisent au sein du réseau électrique à chaque pas de temps grâce à la méthode de résolution itérative présentée en section 2.6.4. Ces flux de puissance permettent ensuite de calculer l'énergie électrique consommée par l'ensemble des trains circulant sur la ligne et de pouvoir ainsi comparer les solutions initiales.

Dans le cas d'une phase transitoire, le mode opératoire diffère sensiblement par la manière de simuler le carrousel transitoire, en effet, il est nécessaire de recréer la régulation de trafic qui s'effectue pour passer d'un carrousel établi à un autre. De fait, selon les cas d'étude, il est nécessaire d'insérer ou de retirer un certain nombre de trains, en suivant les règles de régulation utilisées par l'exploitant de la ligne de métro.

Ensuite, les algorithmes d'optimisation effectuent leurs routines jusqu'à ce que la condition d'arrêt basée sur la stagnation de l'optimum soit atteinte.

3.5.3 Résultats d'optimisation

La figure 3.19 présente une comparaison des puisances électriques moyennes consommées par le carrousel dans le cas de l'utilisation d'une table horaire nominale et dans le cas d'une table horaire optimisée.

La figure 3.20 illustre, quant à elle, l'évolution du gain énergétique global de la ligne de métro au cours de la journée d'exploitation.

Cette optimisation a été réalisée en utilisant la méthode hybride OEP-AG composée d'une cinquantaine d'individus. Le critère d'arrêt retenu porte sur la stagnation du coût de la fonction objectif : si sur un certain nombre d'itérations, l'optimum trouvé n'évolue pas, l'algorithme est déclaré convergent et la procédure d'optimisation est stoppée.

Pour le cas d'étude présenté ici, un gain énergétique journalier de l'ordre de 8% a été enregistré.

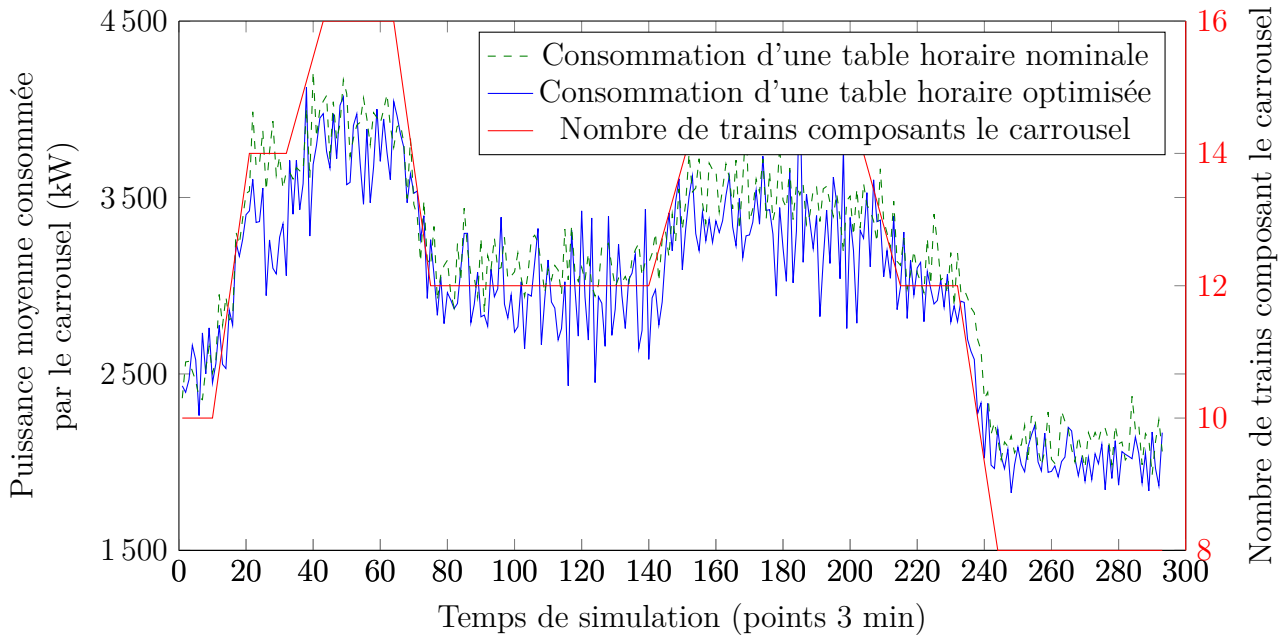


FIGURE 3.19 – Comparaison de puissances électriques consommées par une ligne de métro sur une journée d'exploitation pour deux tables horaires différentes.

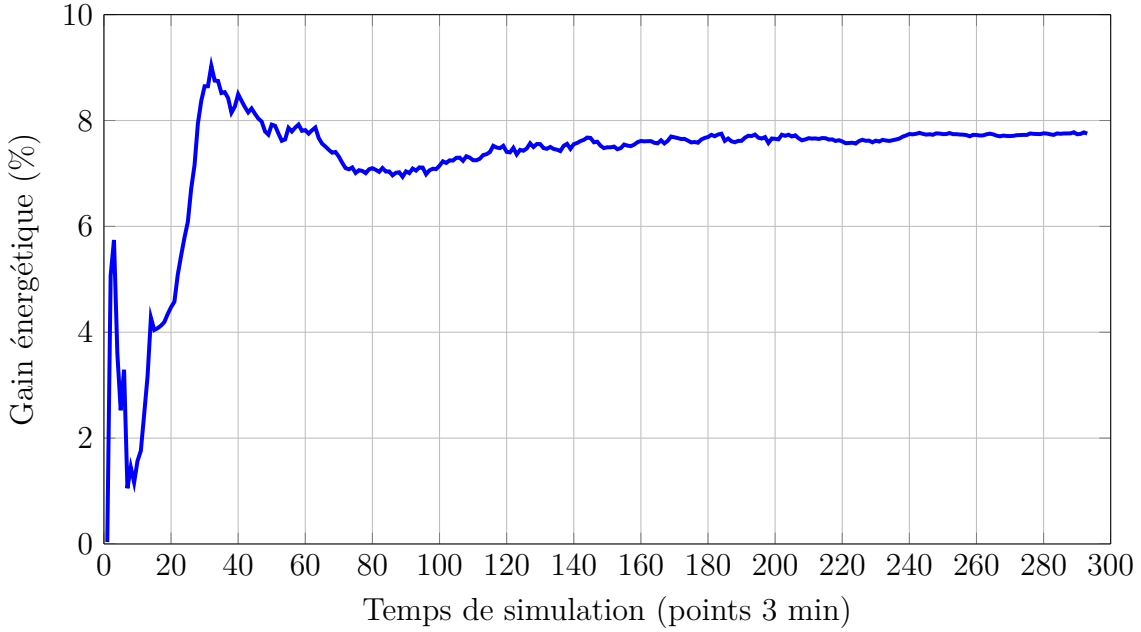


FIGURE 3.20 – Gain énergétique réalisé en exploitant la table horaire optimisée.

3.6 Limites de l'approche hors-ligne

Dans ce chapitre, l'objectif était de présenter des méthodes permettant d'optimiser des phases d'exploitations idéales sans aléas d'exploitation. Cependant, aux vues des performances des méthodes d'optimisation en terme de temps de calcul, l'utilisation d'une telle approche hors-ligne, n'est pas envisageable dans une approche dynamique.

En effet, le temps de calcul nécessaire pour effectuer l'optimisation d'une phase d'exploitation est trop élevée pour une utilisation de ces travaux en temps réel.

3.6.1 Temps de calcul

La contrainte sur le temps de calcul de la boucle d'optimisation est primordiale pour respecter la définition du temps réel.

Une application est ainsi qualifiée de temps réel lorsqu'elle a la capacité d'analyser un certain nombre d'informations (ou événements) issus d'un processus sans perdre un seul de ces événements, ou lorsque son temps de réponse est négligeable par rapport aux phénomènes physiques observés.

Dans le cadre de l'optimisation des temps d'arrêt en station, une mise en œuvre en temps réel signifierait que l'on serait capable de déterminer le temps optimal de stationnement que doit effectuer un train une fois que celui-ci est arrivé en station. La notion de temps réel doit alors s'appliquer avec une granularité de l'ordre de la dizaine de seconde, soit la durée moyenne d'un temps de stationnement nominal.

De fait, la durée de la boucle d'optimisation ne doit pas dépasser cette valeur afin de respecter le critère de temps réel.

Cependant, d'après le tableau 3.4, dans cet intervalle de temps, seule une solution peut être explorée par une méthode AG ou OEP, ce qui n'est ni suffisant pour assurer l'optimalité de la solution explorée ni pour atteindre une convergence.

3.6.2 Optimalité des solutions trouvées

Une autre limite de la méthodologie d'optimisation, développée dans ce chapitre, concerne l'optimalité des solutions trouvées. La méthode hybride OEP-AG permet de concilier les deux aspects fondamentaux d'une technique d'optimisation : la diversification et l'intensification.

Néanmoins, bien que cette méthode semble converger en un nombre fini d'itération, aucune garantie d'optimalité ne peut être apportée pour justifier l'atteinte de l'optimum global de l'espace des solutions. Il serait alors nécessaire de multiplier les essais pour tenter d'améliorer l'optimum trouvé, ce qui n'est pas réalisable avec la puissance de calcul à disposition.

3.7 Conclusion

3.7.1 Résumé des travaux effectués

Après avoir déterminé les paramètres sur lesquels il était possible d'influer afin de réaliser l'objectif de réduction de la consommation énergétique d'une ligne de métro, le problème d'optimisation a été formulé de manière à expliciter les contraintes à respecter et à les intégrer dans la définition de la fonction objectif.

Une première méthodologie d'optimisation de l'intervalle d'exploitation a ensuite été proposée pour déterminer le cadencement optimal d'une ligne de métro dans le cas d'une exploitation sans aléa. Cette première étape d'optimisation révèle que la consommation énergétique d'une ligne de métro peut être fortement réduite en modifiant le cadencement de la ligne de quelques secondes.

La deuxième étape d'optimisation présentée dans ce chapitre concerne la détermination des temps d'arrêt en station permettant de resynchroniser les phases d'accélération et de freinage des trains. De par l'importance de l'espace des solutions à explorer pour résoudre ce problème, l'utilisation d'une optimisation par métaheuristique a été privilégiée pour ses bonnes propriétés de convergence dans des espaces de grandes dimensions.

Les techniques d'optimisation par AG et OEP ont alors été développées pour assurer la résolution du problème, cependant, la nécessité d'assurer le compromis exploration-exploitation en un temps de calcul le plus faible possible a amené à définir une méthode hybride permettant de synthétiser les points forts de l'AG et de l'OEP dans un seul algorithme de recherche.

Moyennant une détérioration des performances en temps de calcul, cette méthode hybride permet alors d'augmenter grandement la densité de solutions jugées acceptables tout en assurant une convergence vers un optimum plus élevé que ceux trouvés par les méthodes AG et OEP.

Puis, l'ensemble de la méthodologie d'optimisation a alors été appliquée à une table horaire journalière et un gain énergétique d'environ 8% a été obtenu en considérant des conditions idéales d'exploitation ne comportant pas d'aléa.

Enfin, dans ce chapitre, l'approche hors-ligne qui a été proposée n'a permis de répondre que partiellement à la problématique initiale de la thèse. En effet, avec cette approche, il est possible d'effectuer une optimisation de la consommation énergétique d'une ligne de métro dans le cas d'une exploitation maitrisée.

Cependant, il est encore nécessaire d'apporter une solution au problème de redéfinition en temps réel des temps d'arrêt en station pour réduire la consommation énergétique tout en intégrant les perturbations de trafic qui se produisent dans les conditions réelles d'exploitation.

3.7.2 Perspectives

Ainsi, des études supplémentaires doivent encore être réalisées afin de répondre aux deux sous-problèmes suivants :

Réduction du temps de calcul de la boucle d'optimisation Il serait nécessaire soit d'augmenter la capacité de l'unité de calcul servant à effectuer l'optimisation, soit de diminuer le nombre d'itérations nécessaires pour effectuer l'optimisation ou alors de synthétiser la méthode d'optimisation par un approximateur universel.

L'augmentation de la capacité de l'unité de calcul n'est pas envisageable pour des raisons économiques et logistiques tandis que la diminution du nombre d'itération n'est pas concevable car cela amènerait à détériorer la qualité des solutions trouvées. En revanche, l'implémentation d'un approximateur universel est parfaitement adaptée pour fournir une aide à la décision en un temps très faible [108].

Optimalité de l'aide à la décision Une première mesure consisterait à procéder à de multiples optimisations partant de points initiaux distincts et une seconde serait d'effectuer un apprentissage de l'ensemble des solutions explorées.

La première solution est en contradiction directe avec l'objectif de réduction du temps de calcul pour effectuer la procédure d'optimisation en temps réel, tandis que la deuxième solution présente l'avantage de pouvoir choisir l'action la plus adaptée au cas d'exploitation étudié.

Il est également à noter qu'une méthode mathématique exacte serait également une option envisageable pour déterminer l'optimum global de l'espace des solutions du problème. Cependant, cette piste n'a pas été étudiée.

Le chapitre suivant s'efforcera donc de concilier les deux contraintes précédentes afin de mettre eu œuvre une méthode capable de fournir une solution optimale dans l'intervalle de temps alloué par le temps de stationnement des trains.

Chapitre 4

Optimisation temps réel des tables horaires

« *Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution. It is, strictly speaking, a real factor in scientific research.* »

Albert Einstein

Sommaire

4.1 Introduction

4.1.1 Limites de l'approche hors-ligne

Les tables horaires sont conçues pour des conditions d'exploitation optimales où aucune perturbation de trafic ne se produit.

Cependant, dans un cas réel d'exploitation les aléas sont inévitables du fait de la présence de facteurs humains qui influent sur le fonctionnement de la ligne de métro automatique.

L'optimisation hors-ligne des paramètres d'exploitation permet de définir un ou plusieurs points de fonctionnement de la ligne jugés comme optimaux d'un point de vue énergétique, mais s'avère inefficace dès lors que le système s'écarte de ces points de fonctionnement.

En pratique, des marges de régulation sont prévues pour assurer la stabilité de l'horaire de passage des trains vis à vis des perturbations mineures qui peuvent être rencontrées. Néanmoins, la régulation n'a pas pour objectif d'assurer un optimum de consommation énergétique et il s'avère alors nécessaire d'insérer de nouvelles règles de fonctionnement pour assurer la réalisation de cet objectif.

4.1.2 Enjeux de l'approche temps réel

L'enjeu de ce chapitre est de définir une méthode pour rendre les travaux présentés précédemment applicables en temps réel, en considérant des conditions réelles d'exploitation intégrant des perturbations de trafic.

Les défis scientifiques et techniques à relever peuvent être synthétisés par la problématique générale suivante : *Comment réaliser une aide à la décision capable de s'adapter aux perturbations d'un système dynamique et de fournir une réponse optimale en temps réel ?* ou en d'autre termes plus spécifiques au sujet de thèse : *Comment effectuer une replanification en temps réel des temps de stationnement des trains pour minimiser la consommation énergétique du carrousel ?*[1]

Nous décrirons donc dans ce chapitre une méthodologie capable de déduire une politique décisionnelle optimale du fonctionnement nominal d'un système, puis de modifier celle-ci pour qu'elle s'adapte aux perturbations rencontrées par ce système.

Concrètement, cela implique de prendre en compte les modifications des conditions de trafic dans la boucle d'optimisation, mais également d'atteindre un temps de calcul pour la boucle d'optimisation qui soit suffisamment faible pour que celle-ci puisse être mise en œuvre en temps réel sur la ligne en exploitation.

Dans la première partie du chapitre, une introduction à l'intelligence artificielle (IA) et au principe de fonctionnement d'un réseau de neurones artificiels (RNA) est réalisée.

Cette introduction présente d'une part nos attentes vis à vis de l'implémentation d'une IA, d'autre part des exemples concrets d'applications qui justifient l'adéquation des RNA pour concrétiser les enjeux visés. Ensuite les différentes méthodes d'apprentissage sont passées en revue pour déterminer celle qui est la plus adaptée pour résoudre

1. Dans ces travaux, toutes les actions de replanification sont assimilées à des modifications de temps d'arrêt en station par rapport aux temps de stationnement nominaux.

la problématique.

La deuxième partie du chapitre est dédiée à la méthodologie de conception d'un estimateur neuronal. L'objectif est de synthétiser la méthode de résolution itérative des flux de puissance dans un RNA. Le RNA serait alors capable d'estimer les flux de puissance qui se produisent sur le réseau électrique entre les trains et les sous-stations en fonction du déplacement des trains et de fournir une approximation de ces flux en un temps très court.

La troisième partie s'intéresse quant à elle à la résolution de la problématique d'optimisation dynamique.
En effet, au chapitre précédent, il a été montré que l'exploration d'une solution du problème d'optimisation des temps de stationnement nécessite environ une dizaine de secondes de temps de calcul, ce qui est incompatible avec un objectif d'optimisation en temps réel. Pour ce faire, le principe d'*apprentissage par renforcement* (AR) est introduit.
Cette méthode permet de déduire une politique décisionnelle d'une suite d'essais et d'erreurs issus d'interactions successives d'un agent apprenant avec son environnement.

Enfin la dernière partie du chapitre est consacrée à l'implémentation de la méthode d'apprentissage par renforcement, et de sa mise en œuvre pour fournir une aide à la décision optimale sur la valeur du temps de stationnement que doit effectuer chaque train pour respecter les contraintes d'exploitation, tout en minimisant la consommation énergétique de la ligne. Une étude des performances de cette méthode est alors effectuée pour en évaluer la capacité à effectuer une optimisation temps réel d'une ligne de métro.

4.1.3 Cahier des charges

Objectifs	• Optimisation temps réel de la consommation énergétique d'une ligne de métro
Contraintes	• Marge de variation des temps d'arrêt en station • Temps de battement • Prise de décision optimale en temps réel • Aléas d'exploitation
Moyens d'action	• Apprentissage des solutions issues d'optimisations
Indicateurs	• Taux de réutilisation du freinage électrique • Déviation par rapport à la table horaire initiale • Temps de calcul

Tableau 4.1 – Cahier des charges de l'optimisation énergétique temps réel d'une ligne de métro automatique

Le cahier des charges de l'optimisation énergétique temps réel est résumé dans le tableau 4.1 et présente les objectifs, les contraintes de l'étude, les moyens d'actions pour réaliser l'optimisation ainsi que les indicateurs utilisés pour évaluer le niveau d'atteinte de l'objectif.

Dans ce chapitre, l'objectif se limite à effectuer une optimisation temps réel des temps de stationnement, car il est considéré que l'intervalle d'exploitation est une contrainte imposée par l'exploitant et que les profils de vitesse ne sont pas des variables d'ajustement puisqu'imposés par la régulation du trafic.

4.1.4 Etat de l'art sur l'optimisation temps réel ferroviaire

Dans le domaine ferroviaire, la notion de replanification en temps réel adopte de nombreuses interprétations. Il convient ici de différencier les Systèmes Légers sur Rails (SLR, dérivé du terme anglais *light rail*), des autres systèmes ferroviaires (comme le fret, les réseaux intercités, ...). La gestion du trafic dans les SLR présente généralement beaucoup moins de contraintes d'exploitation du fait de sa faible longueur et de sa relative simplicité par rapport à un grand réseau ferroviaire présentant des interconnections. Ainsi dans un SLR, des contraintes/actions comme la priorisation des trains aux nœuds d'un réseau, les rotations courtes, le saut de station ou le surstationnement pour réduire le coût opérationnel de transport des passagers ne sont pas mises en œuvre.

Dans cette section, tous les travaux portant sur la replanification temps réel dans le domaine ferroviaire sont traités sans distinction. Cependant il est à noter que certaines actions de replanification mentionnées n'ont pas de raison d'être ou ne peuvent pas être appliquées dans des réseaux ferrés de type métro automatique.

Néanmoins, il reste intéressant d'étudier les procédés explorés dans ces travaux pour traiter le problème de replanification.

Gestion de conflits. Dans [109–112], la replanification temps réel consiste à gérer les conflits d'itinéraires dûs aux aléas d'exploitation pour suivre une table horaire de référence. Des algorithmes d'optimisation sont implémentés pour re-concevoir des tables horaires optimales et robustes après détection d'une perturbation de trafic, afin de continuer à assurer le maximum de connections aux nœuds du réseau ferroviaire tout en réduisant le temps d'attente des usagers.

Respect de la qualité de service et de l'intervalle. [73] tente d'utiliser un système expert à base de logique floue pour effectuer une replanification en temps réel visant à respecter un certain taux de service après aléa [2].

[113] et [114] réalisent une régulation de trafic afin de garantir un intervalle d'exploitation constant entre les trains. Les mesures de régulation ont essentiellement pour objectif d'optimiser les services offerts aux usagers, notamment le temps moyen d'attente.

[115] propose un modèle de contrôle temps réel de l'intervalle d'exploitation par couples de trains consécutifs, visant à modifier les horaires de départs de station pour minimiser la variance de l'espacement temporel entre les trains à chaque station. Néanmoins, avec cette méthode, l'erreur de prédiction peut être propagée et amplifiée au fil de l'optimisation lorsque le nombre de stations augmente. Une approche similaire est également explorée par [116] pour effectuer un contrôle de l'intervalle moyen entre les trains et une minimisation du temps moyen d'attente des passagers, tout en intégrant la notion de réduction de la consommation énergétique de la ligne dans la fonction objectif du problème d'optimisation.

2. Cependant cette technique présente le grand désavantage de nécessiter l'expertise d'un humain pour réaliser des règles floues et des fonctions d'appartenance qui décrivent les contraintes opérationnelles inhérentes à la ligne étudiée.

Dans [117] et [118], Lin propose également de définir des régulations automatiques de trafic pour augmenter la robustesse des tables horaires face aux aléas et assurer la stabilité de l'intervalle d'exploitation par programmation dynamique. Le principe retenu concerne la modification du temps de parcours interstation et la modification des temps de stationnement. Cependant comme l'indique l'auteur dans [119], la résolution par programmation dynamique nécessite de réaliser des recherches vers l'avant qui peuvent engendrer une explosion du temps de calcul. Ainsi, dans [119] et [120], il suggère d'utiliser un processus d'apprentissage par renforcement avec une architecture acteur-critique afin de réaliser les mêmes objectifs que dans ses travaux précédents.

Maximisation de la récupération du freinage électrique. Dans [121], Qu présente un algorithme pour recalculer en temps réel les profils de vitesse optimaux que doivent suivre les trains en interstation pour minimiser la dissipation du freinage électrique, en mettant particulièrement l'accent sur l'analyse de la topographie de la voie pour définir des consignes d'éco-conduites.

[122] et [123] prennent aussi le parti de réduire la consommation énergétique de transports urbains en optimisant les profils de vitesses des trains par alternance de phases de traction, de freinage, de maintien de vitesse et de marche sur l'erre.

Dans [124] et [125], Yin se propose de résoudre le problème de minimisation de la consommation par un processus d'apprentissage par renforcement, en modifiant dynamiquement les profils de vitesse et les temps de stationnement.

Les travaux présentés dans [126] [127] utilisent une approche sensiblement différente : la notion de replanification en temps réel implique qu'aucune table horaire ni aucun intervalle d'exploitation ne sont définis au préalable. L'optimisation a alors pour but de minimiser les coûts opérationnels, le temps de trajet total des passagers ainsi que la consommation énergétique de la ligne. Le principe retenu est la modification des horaires de départs, des temps d'arrêt en station des trains et la modification des profils de vitesse interstation. Dans ces articles, l'auteur fait le choix de proposer des algorithmes qui exploitent de nombreux degrés de liberté pour réaliser un double objectif : satisfaire le client ainsi que l'exploitant.

Parmi les travaux évoqués ci-dessus, ceux traitant d'une replanification temps réel avec un objectif de minimisation de la consommation énergétique souffrent de l'inconvénient majeur de ne pas intégrer de modèle de consommation des trains qui soit fiable ou même représentatif du comportement hautement non linéaire de la caractéristique de renvoi de puissance lors des phases de freinage [3].

3. Lors de cette thèse, en première approche, une modélisation simpliste de la consommation énergétique des trains, avait été employé mais il s'est révélé que l'erreur de simulation commise était comprise entre 20 et 30%. En comparaison, la modélisation énergétique présentée au chapitre 2 entraîne une erreur moyenne de simulation d'environ 6%.

4.1.5 Concept d'intelligence artificielle

Parmi l'ensemble des méthodes d'optimisation existantes, nous avons fait le choix dès l'élaboration du sujet de thèse, de faire appel à des méthodes d'intelligence artificielle (IA).

Ce terme fait référence à des programmes informatiques qui tentent d'imiter le raisonnement humain pour réaliser des tâches complexes. Les comportements adoptés par ces programmes apparaissent alors comme intelligents aux yeux d'un observateur humain.

Au sens strict, le terme *intelligence* implique qu'une machine ou un programme est capable de résoudre des problèmes par une construction originale, sans qu'aucune résolution n'ait été déterminée a priori [128].

Les premières évocations du terme intelligence artificielle remontent aux travaux d'Alan Turing en 1950, dans lequel l'auteur définit un test permettant de déterminer le degré de *conscience* d'une machine : un examinateur juge du type d'interlocuteur à qui il a affaire (humain ou machine) en analysant les réponses fournies à une série de questions.

En 1943, Mc Culloch et Pitts proposent un modèle mathématique pour modéliser le fonctionnement du cerveau : le *neurone formel*. Ils posent ainsi les bases du *réseau neuronal*.

En 1957, Rosenblatt implémente le premier *perceptron*, qui consiste en un réseau de neurones formels composé d'une couche d'entrée et d'une couche de sortie. Ce perceptron a été utilisé pour effectuer de la reconnaissance de forme.

Par la suite, de nombreux projets portés sur le développement d'IA ont vu le jour [4].

Le paradigme qui en résulte est que le problème de conception d'une IA s'est détourné de l'enjeu de l'intelligence vers celui de la *connaissance*.

En effet, dans la pratique, l'implémentation d'une IA se fait par un processus d'apprentissage sur les données disponibles du problème à modéliser. De fait, le degré d'intelligence du système est alors déterminé par la quantité et la qualité des connaissances accumulées lors de l'apprentissage.

En outre, à notre connaissance, seuls [119], [120], [124] et [125] choisissent d'employer des outils issus de l'intelligence artificielle pour réaliser une replanification en temps réel des trains dans un réseau ferroviaire.

A ce titre, [124] et [125] se distinguent singulièrement en offrant une vision intéressante de la replanification adaptée au cas des lignes de métro automatique. La formulation proposée dans [125] est ainsi assez proche de celle proposée dans le chapitre 3 pour réaliser une optimisation hors-ligne des paramètres d'exploitation.

De fait, cette méthode est explorée dans ce chapitre pour rendre le processus d'optimisation hors-ligne réalisable en temps réel.

4. La section 4.2.1 dresse une liste non exhaustive des différents projets qui ont été passés en revue dans le cadre de cette thèse afin de cerner les enjeux concrets de la conception d'une IA efficace.

4.1.6 Nécessité de synthétiser le processus de résolution itératif des flux de puissance

Avant de mettre en œuvre la méthode d'apprentissage pour déduire une politique décisionnelle optimale pour le problème de modification dynamique des temps d'arrêt, il apparait nécessaire de synthétiser la méthode de résolution itérative des flux de puissance sur le réseau.

En effet, dans la section 4.1.4, l'étude des différents travaux portant sur la replanification temps réel dans le domaine ferroviaire a mis en lumière que l'utilisation d'un modèle de consommation énergétique simpliste était privilégié. Cette constatation est conforme à ce qui a été établi au chapitre précédent où il a été montré qu'environ 90% du temps d'exploration d'une solution était consacré à la résolution itérative des flux de puissance entre trains et sous-stations.

Dans le cadre de cette thèse, l'utilisation d'un modèle énergétique simpliste n'aurait pas de sens puisque cela reviendrait à sacrifier la précision du processus d'optimisation au profit d'un gain en temps de calcul. En revanche, la synthèse du processus de résolution itératif par un réseau neuronal permettrait d'annuler en grande partie le temps de calcul nécessaire à l'évaluation de la consommation énergétique liée à une solution.

4.2 Réseaux de neurones artificiels

Le principe des réseaux de neurones artificiels (RNA) est issu de l'analogie entre la biologie et les mathématiques opérée par Mc Culloch et Pitts. Les RNA visent à mimétiser le fonctionnement des neurones à l'intérieur du cerveau humain en propageant les signaux synaptiques et en stockant les informations pertinentes à l'apprentissage d'une tâche donnée.

Un RNA est caractérisé par deux attributs : d'une part, l'organisation des neurones au sein du réseau que l'on appelle architecture et d'autre part l'algorithme d'apprentissage.

L'architecture d'un réseau définit entre autres choses le nombre de neurones composant le RNA et les connexions entre les neurones des différentes couches, à savoir la manière dont le signal est propagé au sein du RNA.

L'algorithme d'apprentissage a pour rôle de stocker le savoir accumulé au sein des coefficients synaptiques du réseau de manière à obtenir le comportement souhaité.

4.2.1 Applications

Les applications du principe de RNA sont multiples dans la littérature :

— La robotique pour le contrôle et le guidage de robots ou de véhicules autonomes [129], [130]

— Les statistiques pour la prévision, la classification et l'analyse de données [131]

— Le traitement du signal pour la reconnaissance de formes et de sons [132], [133]

— La finance pour le calcul de la volatilité d'un marché [134] et la prévision économique de séries temporelles [135]

— Le diagnostique médical [136],[137]

— La création d'intelligence artificielle dans les jeux vidéos [138], [139]
— La gestion de systèmes hydrauliques [140] et des applications en aérospatiale [141]
— Le contrôle de machines électriques [142], la sureté de systèmes électriques [143] et la synthèse de résolution *load flow* [144], [145], [146]
— L'approximation de paramètres de systèmes fortement non-linéaires [147]

Cette liste est loin d'être exhaustive tant les travaux mettant en œuvre des réseaux neuronaux pour des applications spécifiques sont abondants dans la littérature.

La lecture de ces différents travaux fournit un vaste aperçu de ce qu'il est possible de réaliser via la mise en œuvre d'un réseau neuronal, mais permet surtout d'avoir connaissance des obstacles qui ont dû être surmontés pour rendre ce concept mathématique applicable à des cas concrets. A ce titre, ces travaux se révèlent bien plus utiles que ceux portant sur le domaine ferroviaire puisqu'ils permettent d'envisager une approche différente que ce soit sur la formulation, la modélisation ou même la résolution du problème étudié.

4.2.2 Principe des Réseaux de Neurones Artificiels

4.2.2.1 Modèle biologique

En biologie, un neurone est une cellule spécialisée dans le traitement et la transmission d'information dans le cerveau qui constitue l'unité élémentaire du système nerveux. Il se compose généralement d'un corps cellulaire (*péricaryon* ou *soma*) et de prolongements : un axone et des dendrites. Les dendrites sont les ramifications du neurone qui lui permettent de recevoir les signaux électriques et chimiques issus des autres neurones. L'axone est un prolongement de la cellule qui conduit le signal électrique sortant jusqu'aux dendrites auxquelles il est interconnecté. L'échange d'information entre un axone et une dendrite s'effectue par la synapse. L'axone génère un *potentiel d'action* et la synapse assure la conversion et la transmission du signal à la dendrite.

La transmission de l'information neuronale est illustrée par la figure 4.1 [148].

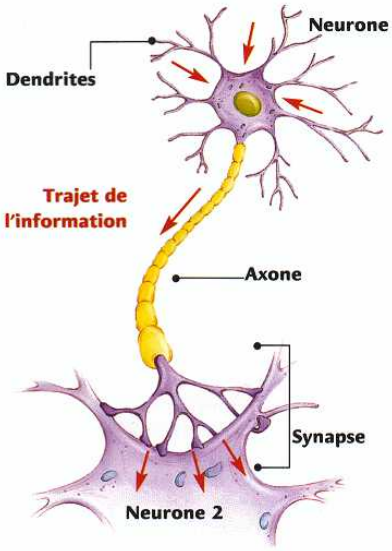


FIGURE 4.1 – Structure typique d'un neurone.

La propagation des signaux électriques ne se fait pas de manière linéaire, mais par un effet de seuil : l'information n'est transmise que lorsqu'un potentiel d'action adéquat est reçu par le neurone.

4.2.2.2 Le neurone formel

Un neurone formel est une fonction algébrique paramétrée de variables réelles dont les valeurs de sorties sont bornées. Un neurone formel est composé de quatre éléments fondamentaux : les entrées, les poids synaptiques associés, une fonction d'agrégation [5] et une fonction d'activation.

Les données d'entrées correspondant aux variables du problème sont pondérées par les poids synaptiques puis sommées et enfin évaluées par une fonction d'activation pour obtenir une sortie. La fonction d'activation permet de recréer l'effet de seuil qui se produit lors de la propagation de l'information dans les synapses. La sortie est la réponse du neurone formel au stimulus reçu en entrée.

Sur la figure 4.2 est représenté le schéma d'un neurone formel possédant n entrées, un biais unitaire et produit une sortie unique notée y.

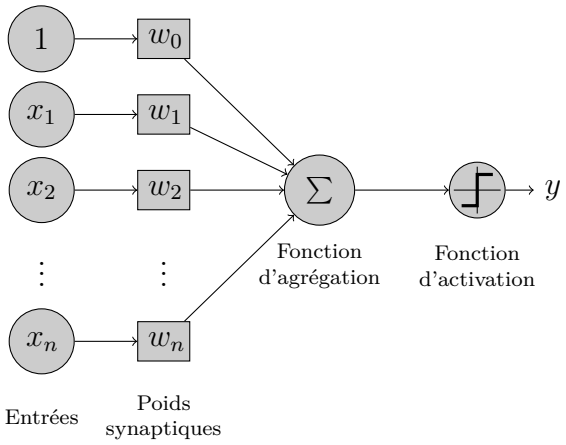


FIGURE 4.2 – Représentation d'un neurone formel à n entrées et 1 sortie.

De fait, le fonctionnement d'un neurone formel est régi par le système d'équations (4.1), où x_i et ω_i représentent respectivement les entrées et les poids synaptiques du neurone. Le biais b intègre une entrée supplémentaire, dont la valeur est fixée à 1, et permet de créer un *offset* pour discriminer les données d'entrées (son rôle est similaire à celui de l'ordonnée à l'origine pour une fonction affine ; ici, $b = 1 \cdot \omega_0$). Le terme Ψ est appelé potentiel d'activation du neurone tandis que ϕ est la fonction d'activation du neurone qui permet de produire la valeur de sortie y.

$$\begin{cases} \Psi = b + \sum_{i=1}^{n} \omega_i x_i = \sum_{i=0}^{n} \omega_i x_i \\ y = \phi(\Psi) \end{cases} \tag{4.1}$$

Les poids synaptiques pondèrent les signaux transmis et régissent le fonctionnement du RNA en fournissant une application de l'espace des entrées vers l'espace des sorties.

5. Dans cette exemple, la fonction d'agrégation est une combinaison linéaire

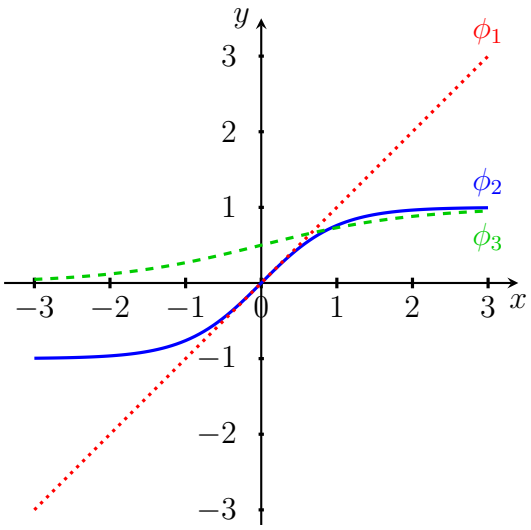


FIGURE 4.3 – Représentation des fonctions d'activation.

Trois types de fonctions d'activation sont classiquement employées.

La fonction identité : $\phi_1(\Psi) = \Psi$

La fonction tangente hyperbolique : $\phi_2(\Psi) = tanh(\Psi)$

La fonction sigmoïde : $\phi_3(\Psi) = \dfrac{1}{1 + e^{-\Psi}}$

La figure 4.3 présente le tracé des fonctions ϕ_1,ϕ_2 et ϕ_3 définies sur l'intervalle [-3,3].

4.2.2.3 Le perceptron multicouche

Un perceptron multicouche (PMC) est une structure composée de plusieurs couches de neurones. Dans un PMC, l'information est propagée de la couche d'entrée vers la couche de sortie. Cette architecture est la plus courante dans la littérature sur les RNA, ainsi, la quasi totalité des travaux cités en section 4.2.1 emploient cette architecture neuronale. L'idée sous-jacente est de combiner plusieurs fonctions élémentaires pour former des fonctions plus complexes.

Un PMC contient une couche d'entrée, une couche de sortie et une (ou plusieurs) couche(s) cachée(s). La couche d'entrée est une couche virtuelle étant donné qu'elle n'a pour rôle que de recevoir les signaux entrants et de les propager aux couches suivantes.

Un perceptron multicouche est alors paramétré par le nombre de neurones composant chacune des couches du réseau, la topologie des connexions entre les neurones (ici, le choix a été fait que tous les neurones d'une couche soient reliés à tous les autres neurones de la couche adjacente), la fonction d'agrégation, l'algorithme d'apprentissage et les fonctions d'activation utilisées par les différentes couches du réseau. Chaque couche du réseau peut ainsi utiliser une fonction d'activation différente afin d'atteindre les objectifs visés.

En outre, il est à noter que hormis les perceptrons multicouches, les réseaux de fonctions à base radiale (RFBR) sont également un type d'approximateur universel populaire. Les RBFR sont en beaucoup de points similaires aux PMC et différent principalement par la manière dont les signaux émanant des couches précédentes sont

combinées. Ainsi, les RFBR utilisent une distance euclidienne en guise de fonction d'agrégation.

Dans cette étude, nous avons fait le choix d'étudier l'approximation par PMC, d'une part pour sa facilité d'implémentation, d'autre part pour sa très bonne capacité à fournir un approximateur robuste dans un grand nombre d'applications solutionnant des problèmes de grandes dimensions, et enfin les algorithmes d'apprentissage sont efficaces pour converger en un nombre raisonnable d'itérations.

La figure 4.4 représente un PMC constitué de quatre entrées, une couche cachée composée de cinq neurones et une couche de sortie à deux dimensions. Dans ce formalisme, chaque cercle correspond à un neurone et les flèches désignent les poids synaptiques. De plus, la couche cachée et la couche d'entrée contiennent également un biais pour permettre au réseau de développer une meilleure capacité de généralisation.

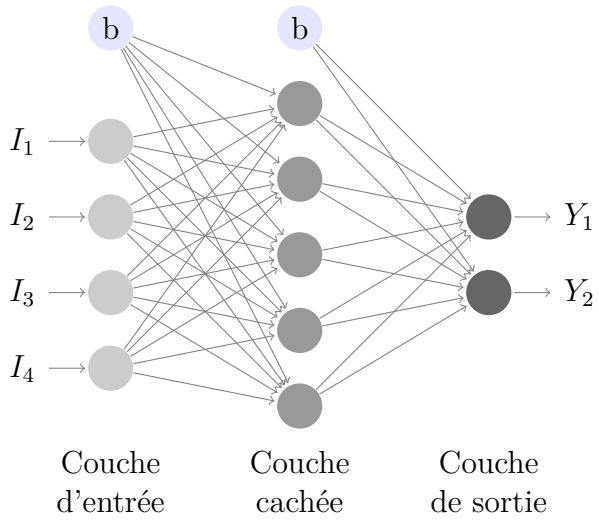


FIGURE 4.4 – Exemple structurel d'un perceptron multicouche.

Pour plus de clarté, les vecteurs contenant les p signaux entrants et les q signaux de sorties sont respectivement notés $I = (I_1, ..., I_p)$ et $Y = (Y_1, ..., Y_q)$.

4.2.3 Notion d'apprentissage

Pour un réseau de neurones artificiels, l'apprentissage d'une tâche ou d'un processus est réalisé par la mise à jour des poids synaptiques. Une phase d'apprentissage consiste donc à modifier ces poids synaptiques jusqu'à ce que le RNA effectue les actions souhaitées.

Les réponses attendues du RNA dépendent alors du problème considéré : dans le cas d'un problème de classification, il peut s'agir de déterminer le centre des classes ou une surface de séparation pour discriminer les cas d'apprentissage, en revanche, dans le cas d'un problème de régression ou d'approximation de fonction le RNA a pour objectif d'approcher une fonction continue sur l'intégralité de son domaine de définition.

Il existe principalement trois types d'apprentissage : l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage par renforcement. Ces apprentissages peuvent être réalisés selon deux modes : en *offline* ou en *online*.

4.2.3.1 Apprentissage supervisé

Lorsque le comportement de la fonction à modéliser est connu, il est possible de constituer une base d'apprentissage composée de couples *entrées-sorties*. Ces couples sont alors utilisés pour effectuer un apprentissage supervisé du réseau neuronal pour l'entraîner à prédire les sorties correspondantes aux entrées. Le réseau s'adapte en modifiant ses poids synaptiques pour converger vers la sortie désirée [149] (figure 4.5).

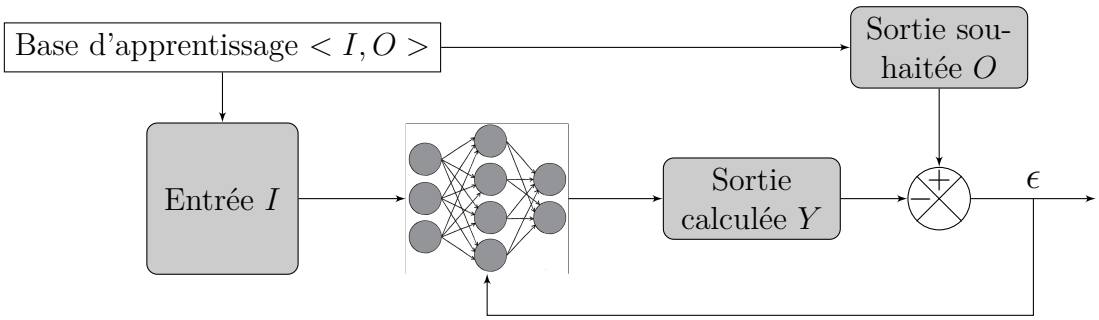


FIGURE 4.5 – Structure de l'apprentissage supervisé.

Le vecteur contenant les sorties souhaitées est formalisé par $O = (O_1, ..., O_q)$, tandis que ϵ représente l'erreur d'estimation commise par le réseau neuronal.

Le défi à relever pour mener à bien un apprentissage supervisé est de calculer l'influence de chaque coefficient synaptique du réseau sur l'erreur d'estimation commise en sortie, puis d'appliquer une règle de modification de ces coefficients pour améliorer le comportement global de l'estimateur neuronal.

4.2.3.2 Apprentissage non-supervisé

Lors d'un apprentissage non-supervisé, le réseau est laissé libre d'évoluer et de converger vers un état final. Les données d'entrées sont présentées et les poids synaptiques sont mis à jour selon une distribution probabiliste : les cas sont classifiés selon leur degré d'appartenance à un sous-ensemble. Le réseau doit déterminer par lui même quelle est la meilleure réponse possible à renvoyer (figure 4.6).

Ce type d'apprentissage est aussi appelé *auto-organisationnel* et est communément utilisé dans des applications de partitionnement, de détections d'anomalies, d'extraction de caractéristiques ou de réduction de dimensions.

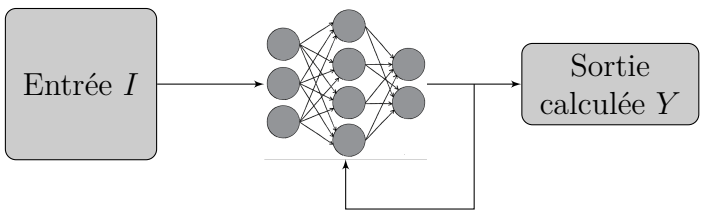


FIGURE 4.6 – Structure de l'apprentissage non-supervisé.

Ce type d'apprentissage est particulièrement employé lorsqu'aucun modèle du système à synthétiser n'est disponible ou que celui-ci est difficilement manipulable. Des exemples typiques d'utilisation seraient : *"A partir de données caractérisant l'évolution*

du courant circulant dans des résistances en fonction de la tension à leurs bornes, comment découvrir la loi d'Ohm ?" ou *"Considérant un ensemble de pages Web, comment classer efficacement ces sites ?"*

Il apparait donc que cette méthode d'apprentissage est plus hasardeuse que l'apprentissage supervisé puisqu'aucune référence n'est utilisée pour orienter la manière dont est réalisé l'apprentissage.

4.2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement (AR) consiste à déduire une stratégie comportementale optimale à partir d'observations de l'état du système. A chaque itération n de l'apprentissage, l'agent apprenant effectue une action a_n depuis un état courant s_n. Cela conduit l'agent à un nouvel état s'_n et à recevoir une récompense r_n pour l'action effectuée.

L'agent apprenant va alors tenter de maximiser les récompenses reçues au cours du temps. La politique de décision est ainsi améliorée itérativement pour atteindre les objectifs de l'approximation (figure 4.7).

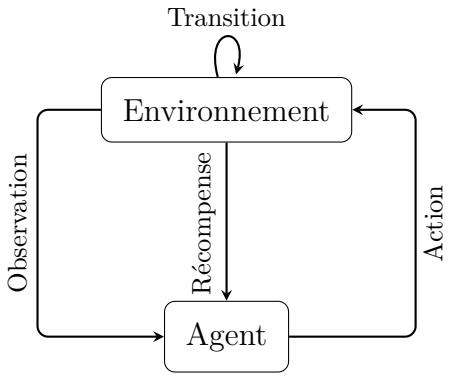


FIGURE 4.7 – Structure de l'apprentissage par renforcement.

En pratique, l'amélioration de la politique se fait par le biais d'une *fonction de valeur*, qui permet de quantifier l'intérêt qu'a l'agent d'effectuer une action a dans l'état s.

L'AR s'avère particulièrement utile pour certains types de problème où l'évolution de l'environnement est incertaine et lorsque les stratégies comportementales efficaces ne sont pas connues.

L'apprentissage par renforcement se distingue de l'apprentissage supervisé par le fait que, la récompense reçue n'indique pas à l'agent si l'action effectuée est optimale.

L'agent doit ainsi effectuer plusieurs actions dans un même état afin de déterminer quelle est la meilleure action possible. Il s'agit donc d'une méthode d'apprentissage par essais et erreurs.

4.2.3.4 Apprentissage *online* ou *offline*

Un apprentissage est qualifié d'incrémental ou online lorsque les données d'apprentissage sont reçues par le système apprenant au fur et à mesure au cours du temps.

A l'inverse, un apprentissage est qualifié de offline lorsque toutes les données nécessaires à l'apprentissage sont connues en amont de l'étape d'apprentissage.

Classiquement, un apprentissage offline ou *batch learning* est composé de deux étapes : une phase d'apprentissage sur les données disponibles puis une phase de test pour évaluer les performances de l'apprentissage sur une nouvelle série de données.

En revanche un apprentissage online adopte une structure itérative : à chaque itération un exemple est reçu par le réseau, puis une estimation est donnée et enfin la performance de prédiction est évaluée pour améliorer les réponses futures.

Le choix d'un apprentissage online ou offline dépend alors surtout de l'application, de la fréquence de disponibilité des données et de la durée d'apprentissage allouée.

4.2.3.5 Choix de la méthode d'apprentissage

Dans la suite des travaux de thèse, nous privilégions l'utilisation de l'apprentissage supervisé pour synthétiser le fonctionnement d'un système. D'une part pour sa simplicité d'implémentation et d'autre part pour la rapidité de convergence, lorsqu'un modèle du système est connu.

Puis lorsque le fonctionnement du système ne peut plus être prédit par un modèle ou est soumis à de trop fortes perturbations, l'utilisation de l'apprentissage par renforcement est envisagé puisque cette méthode permet de réaliser l'apprentissage d'un comportement optimal à partir d'un système de récompense.

L'apprentissage non-supervisé n'est donc pas considéré ici puisque ce type d'apprentissage n'est pas aussi approprié que l'apprentissage supervisé ou que l'apprentissage par renforcement pour synthétiser le fonctionnement d'un réseau ferroviaire en un temps de calcul raisonnable et avec une erreur d'estimation réduite.

4.3 Apprentissage d'un estimateur neuronal des flux de puissance sur un réseau DC

La synthèse d'un processus de résolution par un réseau neuronal a été de multiples fois effectuée dans la littérature. Ainsi, dans [144] , [146], [150], [151] [152] ou [153] un RNA est utilisé pour synthétiser une résolution de type *load flow* dans un système électrique. [140] choisit quant à lui d'effectuer un apprentissage pour contrôler les flux hydrauliques dans une pompe.

Il est intéressant de noter que le dénominateur commun de ces travaux est l'utilisation de l'apprentissage supervisé pour effectuer l'apprentissage du réseau neuronal.

4.3.1 Caractéristiques du problème à estimer

Comme il a été rappelé précédemment, l'apprentissage supervisé est la forme d'apprentissage la plus efficace quand on dispose d'un modèle du système à estimer.

La conception d'un estimateur neuronal se décompose en trois étapes : la création de la base de données, le paramétrage du réseau neuronal et l'implémentation de l'algorithme d'apprentissage.

Une architecture simplifiée de la problématique est résumée par la figure (4.8). Le terme $\Delta\omega_{ij}$ représente la matrice de modification des poids synaptiques.

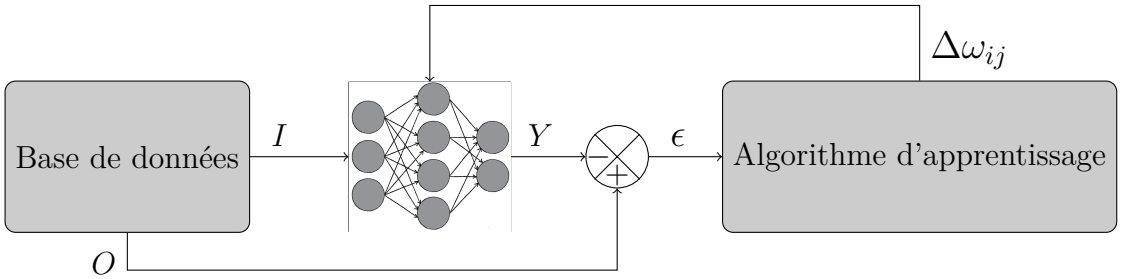


FIGURE 4.8 – Architecture simplifiée de la méthode d'apprentissage supervisé.

4.3.2 Constitution de la base de données

4.3.2.1 Modélisation et simulation des cas d'apprentissage

La constitution de la base de données d'apprentissage se fait par simulation des points de fonctionnement du système que l'on souhaite faire apprendre au réseau neuronal.

Pour cela, il est nécessaire de modéliser puis de simuler le fonctionnement énergétique de la ligne de métro dans différentes conditions d'exploitation afin de caractériser les flux de puissance électrique générés par le déplacement des trains.

Les entrées I du système à synthétiser sont définies comme les positions X_i et les puissances électriques théoriques ${P^{th}}_i$ des i trains composant le carrousel tandis que les sorties O sont les puissances électriques réelles ${P^r}_i$ consommées/renvoyées par les trains : $I = [X_i, {P^{th}}_i]$ et $O = [{P^r}_i]$, $\forall i \in [\![1, N_{trains}]\!]$

4.3.2.2 Segmentation de la base d'apprentissage

Dans le cadre d'un apprentissage supervisé, une base de données est classiquement composée de trois sous-ensembles : une base d'apprentissage, une base de test et une base de validation.

La base d'apprentissage concentre les cas qui seront appris par le réseau neuronal, la base de validation a pour but d'une part de vérifier l'évolution de l'erreur d'apprentissage et d'autre part de s'assurer qu'il n'y ait pas de sur-apprentissage durant le processus et enfin, la base de test est utilisée pour évaluer l'erreur d'estimation commise par le réseau obtenu après apprentissage.

La base de test peut alors avoir comme intérêt de permettre une comparaison objective entre différentes architectures de réseau neuronal ayant appris la même base d'apprentissage.

L'appartenance des données à l'un ou l'autre de ces sous-ensembles est définie aléatoirement pour ne pas biaiser l'estimation et permettre une répartition des données selon une loi normale.

4.3.3 Paramétrage du réseau neuronal

4.3.3.1 Paramétrage de l'apprentissage

La mise en œuvre d'un algorithme de rétropropagation pour l'apprentissage d'un réseau de neurones nécessite de régler un certain nombre de paramètres.

Le nombre de couches : ce paramètre doit être choisi par essais et erreurs puisque l'architecture d'un perceptron est largement dépendante du problème à approcher et des données disponibles dans la base d'apprentissage. [154] a ainsi montré qu'un perceptron possédant deux couches cachées est capable d'approximer n'importe quelle fonction non-linéaire et de constituer des sous-ensembles pour n'importe quel problème de classification.

Le nombre de neurones par couche influence directement la capacité de généralisation du RNA et le temps de calcul nécessaire à la rétropropagation du gradient de l'erreur d'estimation. En effet, plus un réseau possède de neurones plus il est capable de décrire précisément un phénomène, mais parallèlement le temps de calcul augmente puisque chaque neurone d'une couche i impose de devoir calculer N_j coefficients synaptiques et donc N_j opérations de dérivations à chaque itération (où N_j est le nombre de neurones composant la couche suivante).

Les fonctions d'activation : le choix des fonctions d'activation utilisées par chaque couche du RNA est dépendant de l'espace de définition des sorties souhaitées. Ainsi, comme l'explique [155], l'utilisation d'une fonction sigmoïde est très populaire car elle produit des sorties dont la moyenne est nulle. De la même manière, la fonction tangente hyperbolique produit des sorties strictement positives, ce qui peut être une caractéristique importante dans certains cas. Le choix des fonctions d'activation doit donc faire l'objet soit d'une analyse poussée sur les sorties souhaitées au niveau de chaque couche, soit d'un grand nombre d'essais et d'erreurs pour mettre en évidence l'architecture la plus adaptée au problème.

Initialisation de poids synaptiques : généralement, les poids synaptiques sont tirés aléatoirement sur l'intervalle $]-1;1[$, mais il est possible de réduire ce domaine pour ajuster l'apprentissage aux fonctions d'activation choisies. De la même manière que pour le choix des fonctions d'activation, la plage d'initialisation des poids synaptiques est largement dépendante de la nature du problème à synthétiser.

Dans la littérature, malgré les études complètes sur le sujet comme celle de [155], il n'existe aucune règle absolue pour déterminer l'architecture optimale d'un perceptron.

4.3.3.2 Construction et élagage

Une méthode éprouvée pour déterminer la structure optimale d'un PMC consiste à comparer les performances de différentes architectures par essais et erreurs ; cependant, cette méthode présente des limites lorsque la taille de la base de données augmente puisque la durée d'apprentissage croît également.

Dans la littérature, deux méthodes ont été proposées pour déterminer l'architecture optimale d'un réseau de neurones pour réaliser un objectif donné.

Les méthodes de construction. A partir d'une architecture simple, des neurones ou des couches cachées sont ajoutés au cours de l'apprentissage pour accélérer la diminution de l'erreur d'apprentissage [156]. L'architecture du réseau est alors

construite itérativement, les neurones supplémentaires ont pour rôle soit d'améliorer la classification des données [157], soit d'apprendre la corrélation entre l'erreur d'apprentissage et les sorties du réseau [158], ou alors de minimiser l'erreur vue par une couche donnée [159].

Les méthodes d'élagage. L'idée est ici de partir d'un réseau de neurone complexe et de le simplifier au fur et à mesure de l'apprentissage pour améliorer ses capacités de généralisation.

Certains travaux préconisent d'ajouter un terme de pénalité, représentant une mesure de la complexité du réseau, à la fonction objectif ; de sorte que durant l'apprentissage, la simplification du réseau devient un des objectifs de la mise à jour des poids synaptiques [160], [161].

D'autres méthodes ont pour but de simplifier le réseau neuronal en supprimant des poids synaptiques à l'issue de l'apprentissage. L'architecture du réseau neuronal est alors volontairement sur-dimensionnée pour ensuite supprimer des liaisons entre neurones de couches successives. [162] propose de supprimer les liaisons dont les poids synaptiques sont proches de 0. [163] et [164] choisissent quant à eux d'intégrer un terme de sensibilité pour mesurer la variation de la réponse du réseau en fonction de l'évolution de la fonction coût entrainée par la suppression de chaque poids.

Ces méthodes de construction et d'élagage peuvent être assez lourdes et complexes à mettre en œuvre selon le cas d'étude, en nécessitant de déterminer des paramètres de réglage supplémentaires. Celles-ci s'avèrent particulièrement utiles lorsque plusieurs essais d'architecture n'ont pas donné de résultat satisfaisant sur la valeur finale de l'erreur d'apprentissage.

4.3.4 Algorithme de rétropropagation du gradient

Parmi tous les algorithmes de modification des poids synaptiques d'un RNA qui ont été proposés dans la littérature au fil des années, celui de la rétropropagation du gradient de l'erreur est celui qui a été le plus couramment utilisé et qui bénéficie de la plus vaste littérature dédiée [154], [155]. Pour cette raison, son utilisation dans cette thèse a été privilégiée au profit d'autres algorithmes d'apprentissage. En outre, les règles mathématiques sur lesquelles cet algorithme est basé sont intuitives, ce qui permet une prise en main rapide de la méthode.

4.3.4.1 Calcul de l'erreur de propagation

L'algorithme de rétropropagation du gradient de l'erreur consiste à mesurer l'erreur d'estimation commise par le perceptron entre le vecteur des sorties souhaitées O et le vecteur des sorties observées Y, puis à rétropropager le gradient de l'erreur pour ajuster la valeur des poids synaptiques.

Les équations développées dans cette section concernent le cas général d'un perceptron multicouche possédant plusieurs couches cachées.

L'erreur ϵ_q observée par le k-ième neurone de la couche de sortie est (4.2).

$$\epsilon_k = o_k - y_k \tag{4.2}$$

L'erreur globale E vue par la couche de sortie est définie comme la somme des erreurs quadratiques observées (4.3) [6], où le terme n_{dim} correspond à la dimension du vecteur de sortie. Le terme E peut également être perçu comme la fonction coût du problème que l'on cherche à minimiser.

$$E = \frac{1}{2} \sum_{k=1}^{n_{dim-out}} (o_k - y_k)^2 = \frac{1}{2} \sum_{k=1}^{n_{dim}} (\epsilon_k)^2 \tag{4.3}$$

La réponse donnée par un neurone j de la couche courante suite à un signal reçu de la couche précédente i est exprimée par le système (4.4a & 4.4b). Le terme ω_{ij} est le poids synaptique entre le neurone de la couche précédente et celui de la couche courante, n_{dim-i} représente le nombre de neurones composant la couche i et ϕ est la fonction d'activation utilisé par le neurone j.

$$\begin{cases} \Psi_j = \sum_{i=0}^{n_{dim-i}} (\omega_{ij} y_i) & \text{(4.4a)} \\ y_j = \phi(\Psi_j) & \text{(4.4b)} \end{cases}$$

A chaque itération n, le gradient de l'erreur $\frac{\partial E}{\partial \omega_{ij}}$ est calculé pour être propagé de la couche de sortie vers la couche d'entrée afin de modifier les poids synaptiques w_{ij}. La mise à jour des poids synaptiques Δw_{ij} s'effectue alors selon l'équation (4.5), où μ est le taux d'apprentissage du perceptron. Dans cette équation, il est intéressant de noter que la mise à jour se fait dans la direction opposée du gradient afin de se rapprocher du minimum global de la fonction erreur.

$$\omega_{ij}(n) = \omega_{ij}(n) + \Delta\omega_{ij}(n) = \omega_{ij}(n) - \mu \cdot \frac{\partial E}{\partial \omega_{ij}(n)} \tag{4.5}$$

Le développement de l'expression du gradient de l'erreur permet de mettre en évidence les différents termes utiles pour effectuer la mise à jour des poids synaptiques (4.6).

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \Psi_j} \cdot \frac{\partial \Psi_j}{\partial \omega_{ij}} \tag{4.6}$$

Les termes $\frac{\partial y_j}{\partial \Psi_j}$ et $\frac{\partial \Psi_j}{\partial w_{ij}}$ conservent une même expression quelque soit la couche pour laquelle l'erreur est rétropropagée (4.7 & 4.8).

$$\frac{\partial y_j}{\partial \Psi_j} = \frac{\partial \phi(\Psi_j)}{\partial \Psi_j} = \phi'(\Psi_j) \tag{4.7}$$

$$\frac{\partial \Psi_j}{\partial \omega_{ij}} = \frac{\partial (\sum_{i=0}^{n_{dim_i}} (\omega_{ij} y_i))}{\partial \omega_{ij}} = y_i \tag{4.8}$$

Cependant, le terme $\frac{\partial E}{\partial y_j}$ voit son expression être modifiée pour les différentes couches du perceptron, en effet, la valeur de l'erreur E est dépendante de y_j, selon la couche j considérée.

6. Le coefficient 1/2 permet d'effectuer des simplifications dans la suite du développement

4.3.4.2 Cas de la couche de sortie

Dans le cas de la couche de sortie, la rétropropagation de l'erreur a pour objectif de modifier les poids synaptiques entre la dernière couche cachée et la couche de sortie.

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial \epsilon_j} \cdot \frac{\partial \epsilon_j}{\partial y_j} = \epsilon_j \cdot -1 = -\epsilon_j \tag{4.9}$$

L'expression générale du gradient de l'erreur pour la couche de sortie est donnée par (4.10).

$$\frac{\partial E}{\partial \omega_{ij}} = -\epsilon_j \cdot \phi'(\Psi_j) \cdot y_i \tag{4.10}$$

D'après (4.6), le terme de mise à jour $\Delta\omega_{ij}$ est alors définie par (4.11). Cette équation est plus connue sous le nom de règle de Widrow-Hoff ou règle du delta, où δ_j est le *gradient local* (4.12).

$$\Delta\omega_{ij} = \mu \cdot \epsilon_j \cdot \phi'(\Psi_j) \cdot y_i = \mu \cdot \delta_j \cdot y_i \tag{4.11}$$

$$\delta_j = \epsilon_j \cdot \phi'(\Psi_j) \tag{4.12}$$

4.3.4.3 Cas d'une couche cachée

Dans le cas d'une couche cachée, le terme $\frac{\delta E}{\delta y_j}$ désigne la variation de l'erreur observée sur la couche de sortie par rapport à la variation de la sortie de la couche courante (4.13). Les indices j et i désignent toujours respectivement la couche courante et la couche précédente tandis que l'indice k désigne la couche suivante.

$$\begin{aligned}
\frac{\partial E}{\partial y_j} &= \frac{\partial(\frac{1}{2}\sum_{k=1}^{n_{dim-out}}(\epsilon_k)^2)}{\partial y_j} = \sum_{k=1}^{n_{dim-out}}(\epsilon_k \cdot \frac{\partial \epsilon_k}{\partial y_j}) = \sum_{k=1}^{n_{dim-out}}(\epsilon_k \cdot \frac{\partial \epsilon_k}{\partial \Psi_k} \cdot \frac{\partial \Psi_k}{\partial y_j}) \\
&= \sum_{k=1}^{n_{dim-out}}(\epsilon_k \cdot \frac{\partial(o_k - \phi(\Psi_k))}{\partial \Psi_k} \cdot \frac{\partial(\sum_{j=0}^{n_{dim-j}}(\omega_{jk} y_j))}{\partial y_j}) \\
&= \sum_{k=1}^{n_{dim-out}}(\epsilon_k \cdot (-\phi'(\Psi_k)) \cdot \omega_{jk}) \\
&= -\sum_{k=1}^{n_{dim-out}}(\delta_k \cdot \omega_{jk})
\end{aligned} \tag{4.13}$$

En insérant (4.13) dans (4.6), le terme $\frac{\partial E}{\partial \omega_{ij}}$ devient (4.14).

$$\frac{\partial E}{\partial \omega_{ij}} = -[\sum_{k=1}^{n_{dim-out}}(\delta_k \cdot \omega_{jk})] \cdot \phi'(\Psi_j) \cdot y_i \tag{4.14}$$

L'expression finale de la mise à jour $\Delta\omega_{ij}$ dans le cas d'une couche cachée est similaire à l'expression finale de l'équation (4.11), à la différence que le gradient local δ_j est défini comme (4.15).

$$\delta_j = [\sum_{k=1}^{n_{dim-out}} (\delta_k \cdot \omega_{jk})] \cdot \phi'(\Psi_j) \tag{4.15}$$

4.3.4.4 Taux d'apprentissage et coefficient d'inertie

En plus du taux d'apprentissage μ, qui a pour but de contrôler l'amplitude de la mise à jour des poids synaptiques, un autre paramètre est couramment ajouté pour aider l'algorithme à s'extirper des extrema locaux : le coefficient d'inertie (ou *momentum*) noté α. Ce terme a pour objectif d'empêcher la fonction erreur E de se stabiliser dans un minimum local en ajoutant au terme de mise à jour $\Delta\omega_{ij}(n)$ une fraction de la mise à jour précédente $\Delta\omega_{ij}(n-1)$.

Le terme $\Delta\omega_{ij}$ à l'itération n se définit alors par l'équation (4.16).

$$\Delta\omega_{ij}(n) = -\mu \cdot \frac{\delta E}{\delta\omega_{ij}(n)} + \alpha \cdot \Delta\omega_{ij}(n-1) \tag{4.16}$$

Le taux d'apprentissage et le facteur d'inertie doivent faire l'objet d'une étude attentive pour assurer une diminution de l'erreur de prédiction au fil des itérations. D'après LeCun [155] et [165], une méthode efficace pour dimensionner ces termes est de les rendre dépendants de l'erreur globale du réseau. Ainsi, lorsque l'erreur de prédiction devient faible, le réseau semble converger vers une réponse stable. Le taux d'apprentissage et le terme d'inertie doivent alors avoir une valeur relativement faible pour ne pas perturber la convergence.

4.3.4.5 Normalisation des données

La normalisation des données d'apprentissage n'est pas une nécessité, mais il a été remarqué qu'une convergence plus rapide était obtenue dans les applications où les données d'entrées et de sorties sont normalisées [155], [166]. Ici, la normalisation désigne l'action de réduire la taille de l'espace des données en la ramenant sur un intervalle $]0;1[$ ou $]-1;1[$. Certains auteurs font aussi le choix de resserrer encore l'espace de normalisation en se ramenant à l'intervalle $]0,1;0,9[$. Cependant chaque problème étant unique, il convient de choisir l'intervalle de normalisation le plus approprié afin de réduire le temps de convergence de l'apprentissage.

Plusieurs raisons expliquent l'intêret d'effectuer une normalisation des données :

— La normalisation des données d'entrées permet de donner une importance égale à chacune des composantes du vecteur d'entrée sans tenir compte des valeurs réelles de ces composantes, ni de leurs unités respectives.

— La normalisation permet également de diminuer la difficulté du calcul numérique (ici, la rétropropagation de l'erreur) en permettant un conditionnement faible du problème d'apprentissage. Cette notion de conditionnement fait appel à des théories mathématiques qui ne seront pas explicitées ici, mais le lecteur intéressé peut se référer aux travaux de [167] qui étudie précisément l'impact du conditionnement sur l'apprentissage des réseaux de neurones, mais également sur l'influence d'autres facteurs comme le taux d'apprentisage ou le momentum.

— La normalisation des données permet également d'éviter les effets de saturation : lorsque les signaux reçus par les neurones sont très supérieurs à 1, les fonctions d'activations sont saturées et transmettent en sortie une valeur proche de leurs limites de variations. Ainsi, pour des fonctions d'activation de type sigmoide, ou hyperbolique, dès que la combinaison linéaire des entrées est supérieure en valeur absolue à 3, la sortie renvoyée est proche de 1 en valeur absolue. Ainsi, pour toute valeur supérieure en entrée, la réponse ne variera que très peu.

— Dans le cas des problèmes de régression, [155] précise qu'il est généralement utile de centrer les données d'apprentissage autour de 0 et d'effectuer une normalisation permettant d'obtenir une covariance [7] de 1 pour toutes les données. La mise à l'échelle permet alors d'équilibrer la manière dont l'algorithme de rétropropagation va modifier les coefficients synaptiques connectés à la couche d'entrée.

Le choix de l'intervalle de normalisation dépend également du problème traité. Seul un paramétrage par essais et erreurs peut permettre de déterminer la meilleure plage de variation de la normalisation. Cependant, une méthode assez simple pour avoir une idée de l'intervalle de normalisation est de déterminer quel intervalle est le plus susceptible de faire converger l'apprentissage en fonction du nombre de données d'entrée, de la valeur des sorties souhaitée, du nombre de couches cachées et des fonctions d'activation sur chaque couche. En d'autres termes, il est nécessaire d'évaluer le processus de propagation de l'information dans le réseau avant de choisir la méthode de normalisation.

4.3.4.6 Définition de l'erreur d'apprentissage

L'évaluation des performances d'estimation du RNA sur la base d'évaluation requiert de mettre en place plusieurs types d'erreurs d'estimation afin de suivre l'évolution de l'apprentissage et de s'assurer que celui-ci converge correctement vers l'erreur minimale admise E_{min}. [168]

La racine de l'erreur quadratique moyenne notée $RMSE$ (4.17) est la plus courante des erreurs utilisées, où N_{cas} est le nombre de cas composant la base d'évaluation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{N_{cas}}(O_k - Y_k)^2} \tag{4.17}$$

L'erreur quadratique relative notée RSE (4.18) compare l'erreur quadratique par rapport à la variance de l'estimation, il est possible de trouver cette erreur sous le nom d'erreur quadratique moyenne normalisée puisque le dénominateur est égal au produit de la variance estimée de la série σ^2 par le nombre de cas d'apprentissage dans la base de données. Ce type d'erreur présente l'avantage que sa valeur n'augmente pas proportionnellement à la taille de la base de données. En outre, la normalisation par la variance permet de comparer des séries d'amplitudes différentes [169]. Le terme $\overline{Y_k}$ représente la moyenne des sorties du réseau neuronal pour le k-ième cas d'apprentissage.

$$RSE = \frac{\sum_{k=1}^{N_{cas}}(O_k - Y_k)^2}{\sum_{k=1}^{N_{cas}}(\overline{Y_k} - Y_k)^2} = \frac{\sum_{k=1}^{N_{cas}}(O_k - Y_k)^2}{N_{cas}\,\sigma^2} \tag{4.18}$$

7. L'utilisation d'une covariance permet de spécifier la dispersion des données : c'est une mesure qui caractérise l'écart entre les données par rapport à leurs espérances mathématiques respectives. La covariance étudie donc la variation des variables entre elles.

L'erreur absolue moyenne notée MAE (4.19) offre un indicateur sur la déviation moyenne de l'erreur d'estimation.

$$MAE = \frac{1}{n} \sum_{k=1}^{N_{cas}} |O_k - Y_k| \tag{4.19}$$

Le taux d'erreur moyen noté MPE (4.20) intégre un facteur de mise à l'échelle afin de pouvoir comparer différentes prédictions quelque soit le point de fonctionnement du système.

$$MPE = \frac{1}{n} \sum_{k=1}^{N_{cas}} \frac{Y_k - O_k}{Y_k} \tag{4.20}$$

Généralement, il est supposé que le bruit entachant les données est supposé suivre une distribution gaussienne. Dans le cadre de cette thèse, nous ne disposons pas des outils mathématiques pour effectuer la démonstration d'une telle hypothèse. De fait, un autre type d'erreur est introduit : l'erreur Laplacienne normalisée notée NLE (4.21). Comme son nom l'indique, la NLE présume que le bruit des données suit une distribution laplacienne et de la même manière que l'erreur RSE, la NLE subit une mise à l'échelle afin que sa valeur ne subisse pas l'influence de l'augmentation de la base de données.

$$NLE = \sum_{k=1}^{N_{cas}} \frac{|Y_k - O_k|}{|\overline{Y_k} - Y_k|} \tag{4.21}$$

Dans [168], l'auteur implémente les erreurs MSE, RSE, MAE et MPE afin de comparer les performances de cinq types de RNA ayant appris la même base de données.

Le suivi de l'évolution de ces erreurs permet alors entre autres choses de savoir s'il est nécessaire de modifier le paramétrage du RNA ou d'arrêter le processus en cas de stagnation des erreurs d'estimation.

4.3.4.7 Implémentation de l'algorithme

L'algorithme de rétropropagation, dans la version basique d'un apprentissage supervisé, est composé des étapes présentées dans l'algorithme 8.

Algorithme 8 Algorithme de rétropropagation du gradient de l'erreur

Entrée(s) Base d'apprentissage composée de N_{cas}-couples (I_n,O_n)
Initialiser la structure du PMC, les coefficients synatiques ω_{ij} de toutes les couches et définir les fonctions d'activation de chaque couche.
Tant que E est supérieur au critère d'arrêt **Faire**
Pour n = 1 : N_{cas} **Faire**
Propager le signal d'entrée I_n sur les couches du PMC (4.4a & 4.4b)
Calculer la sotie du réseau (4.4b pour la couche de sortie)
Calculer l'erreur d'estimation par l'équation (4.3)
Mettre à jour les poids synaptiques par l'équation (4.16)
Fin du Pour
Fin du Tant que
Sortie(s) Coefficients synatiques entrainés ω_{ij}

La mise en œuvre de l'algorithme 8 nécessite de définir certains paramètres :

Le critère minimal d'estimation E_{min}. L'algorithme de rétropropagation doit itérer la modification des poids synaptiques jusqu'à ce que le coût de la fonction E converge vers E_{min}. Ce critère est choisi de manière à arrêter l'apprentissage lorsqu'une précision suffisante est atteinte par le perceptron. En outre, il est également nécessaire de vérifier qu'au cours de l'apprentissage, les différentes erreurs d'estimation décroissent constamment. Une stagnation de ces erreurs sur un trop grand nombre d'itérations amène le RNA à faire du sur-apprentissage (*overfitting* en anglais) ce qui a pour conséquence principale de lui faire perdre ses capacités de généralisation. En outre, un sur-apprentissage peut aussi survenir lorsque la taille du réseau de neurones est trop conséquente par rapport à la taille de la base de données à apprendre : le RNA se comporte alors comme une table stockant les cas d'apprentissage.

La taille de la base d'apprentissage. Une base d'apprentissage de taille importante impose une phase d'apprentissage longue afin de développer les capacités de généralisation du RNA. Mais plus la base d'apprentissage est grande et plus l'estimation du système est précise.

La méthode d'apprentissage. Lors de l'apprentissage deux modes d'apprentissage peuvent être envisagés : l'apprentisssage offline ou l'apprentissage online. Le premier consiste à présenter successivement tous les cas de la base d'apprentissage au réseau de neurone afin de calculer une erreur moyenne à rétropropager alors que dans le second, un cas d'apprentissage est sélectionné aléatoirement à chaque itération, et l'erreur commise sur l'estimation de ce cas d'apprentissage est utilisée pour mettre à jour les coefficients synaptiques. L'apprentissage stochastique en ligne est très souvent privilégié pour ses propriétés de convergence plus élevées et sa meilleure capacité à suivre l'évolution de l'erreur de prédiction [170], [155].

Taux d'apprentissage μ et momentum α. Ces deux paramètres sont à choisir par essais et erreurs pour définir le couple (μ, α) qui permet d'effectuer un apprentissage efficace le plus rapidement possible.

4.3.4.8 Performances de l'estimation

La base d'apprentissage permet au RNA de développer des capacités d'estimation et de généralisation pour fournir la sortie la plus proche de la réponse attendue grâce à l'algorithme de rétropropagation du gradient, puis la base de validation est utilisée pour suivre l'évolution de l'erreur d'apprentissage et ainsi prendre les décisions adéquates sur la modification des paramètres de l'apprentissage ou la poursuite/arrêt du processus d'apprentissage.

Lorsque les erreurs définies en section 4.3.4.6 ont atteint les seuils souhaités, il est nécessaire d'effectuer une dernière validation des performances d'estimation du RNA en évaluant son aptitude à fournir une estimation sur la base de test. Cette dernière étape est indispensable pour certifier que le réseau est capable de fournir une estimation fiable pour des points de fonctionnement non appris.

Dans ce contexte, deux critères sont classiquement utilisés : le premier est le coefficient de corrélation linéaire (CCL) noté R, appelé aussi coefficient de Bravais-Pearson, qui permet de mesurer la qualité de l'estimation. Le CCL entre les sorties souhaitées et les sorties produites par le réseau neuronal est défini par le rapport entre la covariance

des données et le produit de leurs écart-types (4.22).

$$R = \frac{\sum_{k=1}^{n}(o_k - \overline{o})(t_k - \overline{t})}{\sqrt{\sum_{k=1}^{n}(o_k - \overline{o})^2 \sum_{k=1}^{n}(t_k - \overline{t})^2}} \tag{4.22}$$

Ce coefficient est donc une mesure de l'intensité de la liaison linéaire unissant les variables estimées et calculées. Le CCL est un nombre adimensionel défini sur l'intervalle $[-1; 1]$. Un CCL de valeur nulle indique que les séries O et Y sont linéairement indépendantes, tandis qu'un coefficient proche de l'unité indique une forte corrélation. Un CCL négatif révèle que Y varie en sens inverse de O et inversement en cas de CCL positif.

Le deuxième critère R^2 est le coefficient de détermination (CD), il est défini par (4.23), où le numérateur représente l'erreur quadratique de l'estimation tandis que le dénominateur représente la variance des données. La formulation du CD est ainsi fonction de l'erreur quadratique relative (RSE).

$$R^2 = 1 - \frac{\sum_{k=1}^{N_{cas}}(O_k - Y_k)^2}{\sum_{k=1}^{N_{cas}}(\overline{Y_k} - Y_k)^2} = 1 - RSE \tag{4.23}$$

Ce critère a pour objectif de donner une indication sur l'erreur d'estimation commise, mais également sur la dispersion des valeurs estimées [171]. Pour un problème de régression, une valeur de R^2 très proche de 1 est souhaitée à l'issue de l'apprentissage, autrement dit le terme d'erreur doit tendre vers 0 ($RSE \approx 0$).

Dans la pratique, ce critère est une sécurité supplémentaire qui n'a pas réellement d'intérêt sous l'hypothèse d'un suivi adéquat de la décroissance des erreurs d'estimation commises sur la base de validation. Cependant, il s'agit d'un critère universel de qualité de la régression qui permet de comparer les performances de convergence de divers algorithmes d'apprentissage.

4.3.5 Description des cas d'étude

Les performances de la méthodologie de conception d'un estimateur neuronal sont évaluées selon deux aspects : la précision de l'estimation et la rapidité de l'estimation.

Pour cela, l'étude des performances de l'estimateur est effectuée sur deux cas d'étude. Le premier cas d'étude consiste à estimer les puissances électriques vues par un carrousel de 16 trains sur un tour de boucle à un intervalle donné. Le second cas d'étude est un peu plus complexe et consiste à estimer les puissances électriques vues par un carrousel de 16 trains sur un tour de boucle pour l'ensemble des intervalles d'exploitation possibles : un carrousel établi de 16 trains est d'abord modélisé puis le modèle énergétique lié au déplacement des trains est résolu itérativement pour connaitre à chaque pas de temps les flux de puissance aux bornes des trains. Le processus est répété pour simuler le fonctionnement du carrousel pour les différents intervalles possibles. Les données issues de ces simulations sont ensuite présentées au réseau neuronal via un apprentissage supervisé. Un pas d'échantillonnage de 1s est utilisé, ce qui donne

une base de données constituée d'environ 3000 cas d'apprentissage pour chaque intervalle possible.

Le second cas d'étude est composé d'environ 12 fois plus de données que le premier cas d'étude, ce qui permet d'effectuer une comparaison de la méthodologie proposée pour différentes tailles de base de données.

Dans ces deux cas d'études, la base d'apprentissage est composée de 70% des données, tandis que les bases de validation et de test sont composées respectivement de 20 et 10% des données. Ainsi, les performances de l'estimation sont évaluées en ne connaissant que 70% des points de fonctionnement du système, ce qui permet de vérifier la capacité de généralisation du RNA.

Il est à noter que la rapidité de l'estimateur concerne le temps de calcul nécessaire à l'estimation d'un certain nombre de points de fonctionnement et non le temps d'apprentissage. En effet, plus le problème à approximer est complexe, plus il sera nécessaire d'allouer un temps d'apprentissage conséquent afin que l'estimateur puisse converger vers le taux d'erreur E_{min} souhaité.

En plus du critère d'arrêt E_{min}, un critère supplémentaire basé sur la valeur du coefficient de corrélation est défini afin que les bases de validation et de test jouent un rôle dans la décision de stopper l'apprentissage.

4.3.6 Performances de l'estimateur neuronal

4.3.6.1 Précision de l'estimation

4.3.6.1.1 Évolution des erreurs d'apprentissage sur la base de validation

Au fil des expérimentations, trois erreurs se sont avérées utiles pour suivre l'évolution de l'erreur d'apprentissage : RMSE, RSE et NLE (figures 4.9 et 4.10).

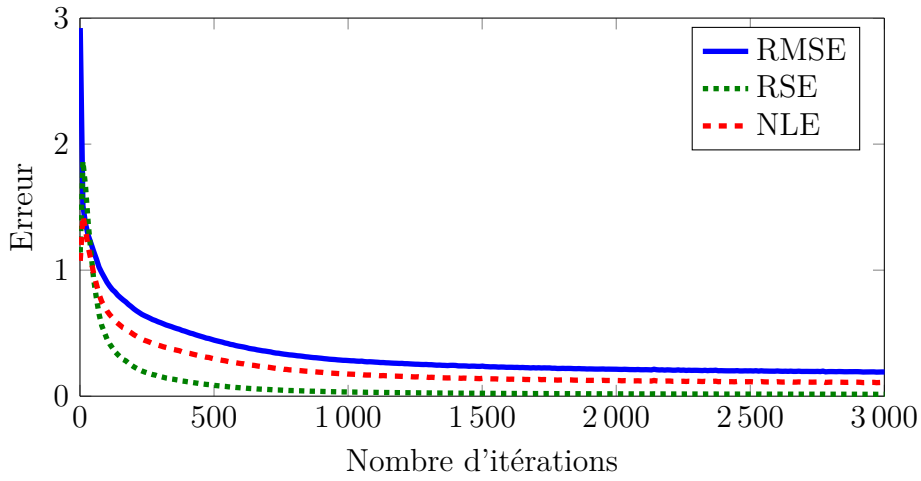


FIGURE 4.9 – Évolution des erreurs d'estimation sur la base de validation (cas 1).

L'erreur MAE a un comportement plus erratique au fil des itérations puisqu'elle représente une moyenne absolue des écarts d'estimation, cependant sa valeur finale à

l'issue de l'apprentissage est une bonne indication de la précision globale de l'estimation sur l'ensemble des points de fonctionnement décrits dans la base de validation.

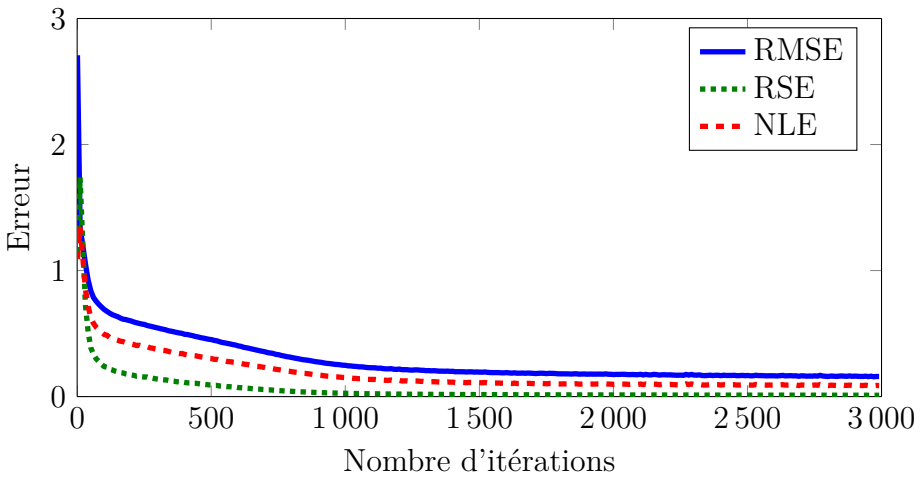


FIGURE 4.10 – Évolution des erreurs d'estimation sur la base de validation (cas 2).

La décroissance continue des erreurs sur les figures 4.9 et 4.10 est une première indication de la validité de l'architecture neuronale et du paramétrage de l'algorithme de rétropropagation.

De ce point de vue, seules les premières itérations présentent un intérêt puisque à partir d'un certain moment l'évolution des erreurs d'estimation n'est plus perceptible.

Ainsi, la première phase de décroissance rapide peut être assimilée à la phase de généralisation tandis que la seconde phase peut être vue comme une phase de spécialisation.

En revanche, l'erreur MPE est inadaptée pour suivre l'évolution de l'erreur d'approximation. En effet, pour plusieurs cas d'apprentissage, la sortie attendue est proche d'une valeur nulle ce qui engendre une MPE très élevée. De fait, cette erreur n'est pas considérée dans le reste de l'étude.

4.3.6.1.2 Évolution des coefficients de corrélation et de détermination sur la base de test

L'évolution du coefficient de corrélation et du coefficient de détermination pour les deux cas d'études sont exprimés par les figures 4.11 et 4.12.

La convergence des coefficients semble plus rapide dans le deuxième cas d'étude. L'une des raisons est que la base d'apprentissage du deuxième cas d'étude est composée de 12 fois plus de données que le premier cas d'étude.

A chaque itération le réseau neuronal ajuste ses poids synaptiques pour adapter son estimation à un plus grand nombre de points de fonctionnement, ce qui entraîne une vitesse initiale de convergence plus élevée.

En revanche, la convergence finale vers une valeur proche de 1 est beaucoup plus longue à obtenir pour les mêmes raisons.

L'allure de ces courbes renseigne également sur l'adéquation de l'architecture du réseau neuronal et des paramètres de l'algorithme de rétropropagation avec le problème à estimer.

— Une évolution plus irrégulière de ces courbes aurait ainsi mis en évidence que le taux d'apprentissage et le momentum ont des valeurs trop élevées ce qui ne permet pas à l'algorithme de rétropropagation de converger dans de bonnes conditions.

— Une convergence des coefficients vers une valeur très inférieure à 1 aurait signifié que l'architecture du RNA n'est pas suffisante pour développer des propriétés de généralisation. De même, une lente convergence peut également signifier que l'architecture du RNA est surdimensionnée.

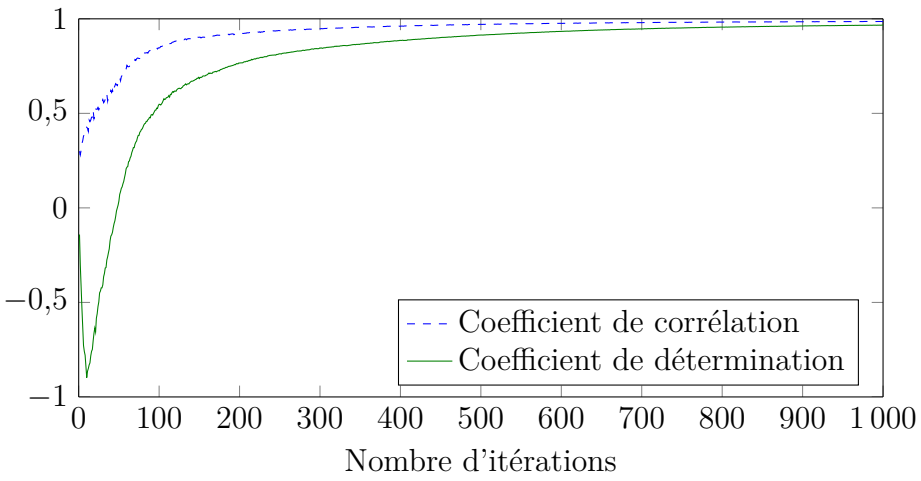


FIGURE 4.11 – Évolution de la corrélation entre données estimées et calculées (cas 1).

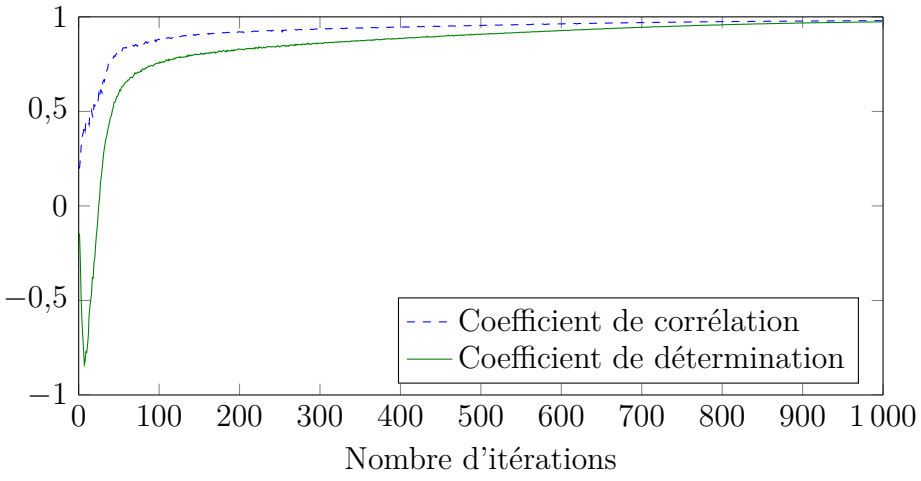


FIGURE 4.12 – Évolution de la corrélation entre données estimées et calculées (cas 2).

4.3.6.1.3 Représentativité de la base de test

A l'issue de l'apprentissage, le coefficient de corrélation et le coefficient de détermination issus de l'évaluation de la base de test ont une valeur très proche de 1, ce qui implique que le RNA a développé des capacités de généralisation permettant de donner une bonne estimation sur de nouveaux points de fonctionnement.

Les figures 4.13 et 4.14 représentent une analyse graphique de la corrélation sur l'ensemble de la base de données initiale.

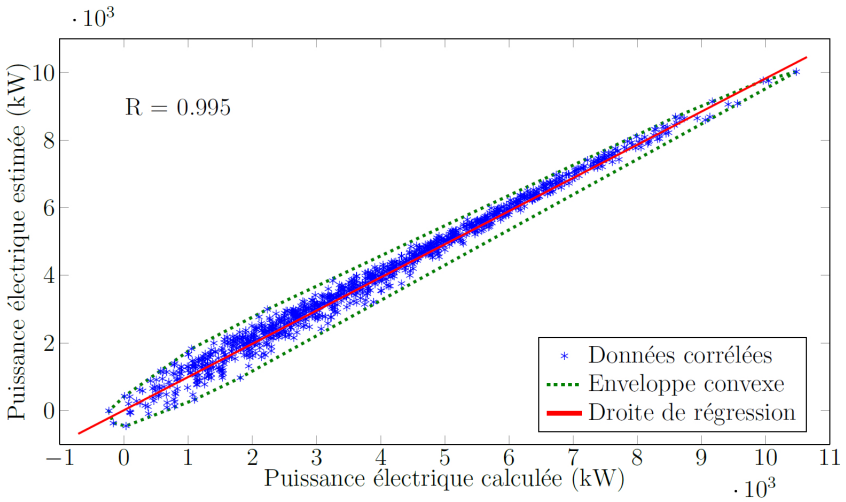


FIGURE 4.13 – Corrélation entre données estimées et calculées (cas 1).

L'abscisse correspond aux données calculées O et l'ordonnée aux données estimées Y. L'intérêt de ces graphiques est de visualiser les performances d'estimation sur l'ensemble des points de fonctionnement du système.

Dans les deux cas d'étude, l'apprentissage a été effectué jusqu'à ce que le coefficient de corrélation atteigne une valeur de 0,995.

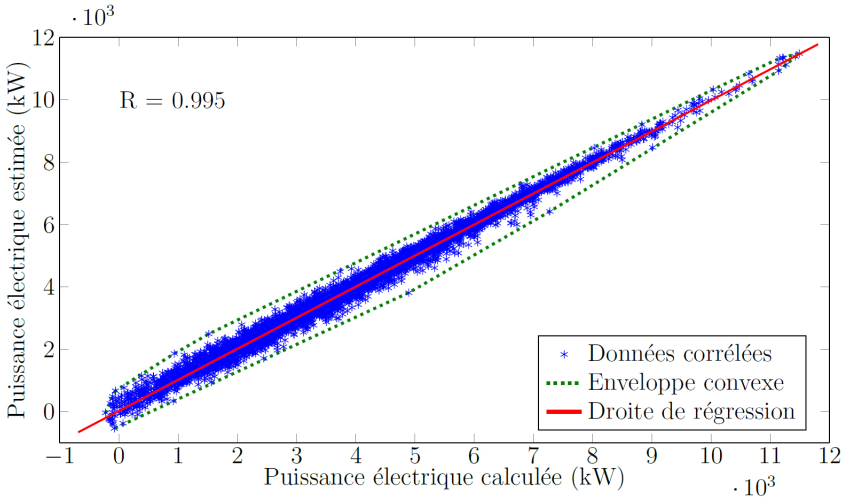


FIGURE 4.14 – Corrélation entre données estimées et calculées (cas 2).

Sur ces figures, l'enveloppe convexe représente la dispersion entre les données calculées et estimées. En théorie, un estimateur parfait verrait l'enveloppe convexe être confondue avec la droite de régression de pente unitaire.

Ainsi, la surface décrite par l'enveloppe convexe est définie comme un critère de dimensionnement supplémentaire pour choisir l'architecture du réseau neuronal la plus adaptée au problème.

Un sous-objectif de la méthodologie pourrait alors être de déterminer l'architecture neuronale et les paramètres de l'algorithme d'apprentissage permettant de minimiser l'aire de l'enveloppe convexe.

4.3.6.1.4 Visualisation de l'erreur d'apprentissage

Les figures 4.15 et 4.16 proposent une comparaison entre données calculées et estimées pour les 500 premières secondes de fonctionnement de la base de données pour les deux cas d'études. Elles constituent une illustration objective de la précision des estimateurs.

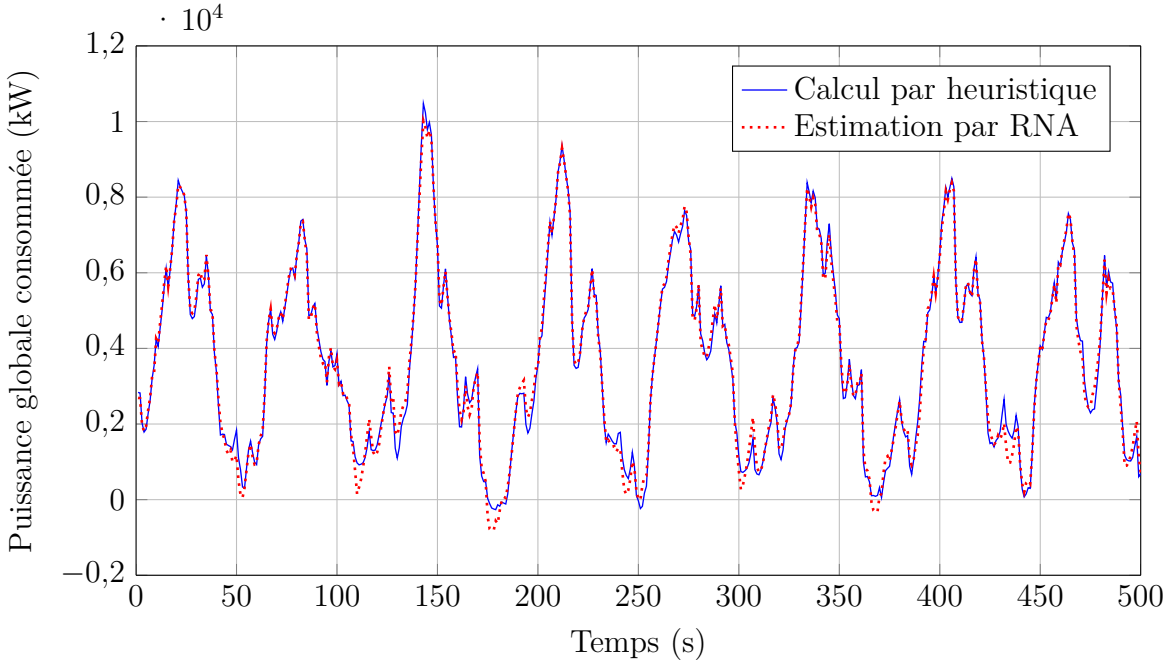


FIGURE 4.15 – Comparaison entre les données estimées et calculées (cas 1).

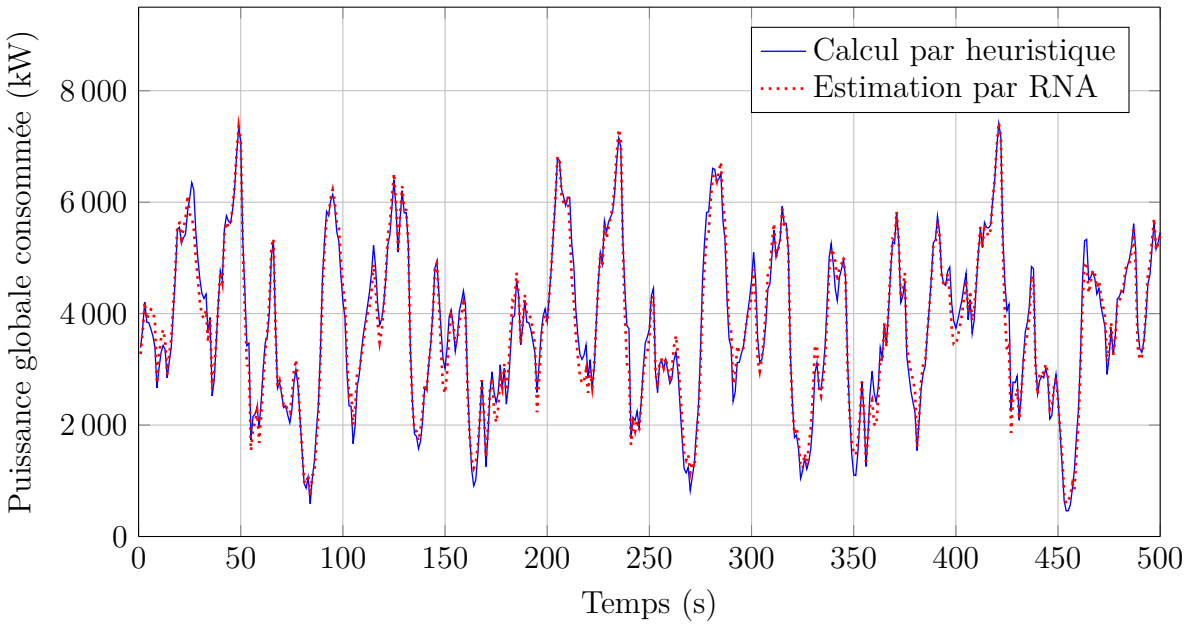


FIGURE 4.16 – Comparaison entre les données estimées et calculées (cas 2).

Ces figures sont obtenues en sommant les puissances électriques aux bornes des trains. Logiquement, la puissance globale résultante devrait être strictement positive quelles que soient les conditions d'exploitation des carrousels, cependant la résolution

itérative permet de calculer les flux de puissance vus par les trains avec une certaine erreur, ainsi la sommation des données calculées et estimées peut être légèrement négative du fait de cette marge d'erreur.

En théorie, le critère d'arrêt de la résolution itérative devrait donc être affiné pour réduire l'erreur commise, cependant le gain en précision pour la méthodologie globale serait nul puisque comme il est montré par la suite, une consommation énergétique du carrousel faible voire négative transmet la même information : la synchronisation des trains est optimale.

En effet, l'objectif final de la méthodologie n'est pas de déterminer précisément la consommation énergétique de la ligne, mais de définir dans quelles conditions d'exploitation cette consommation est minimisée. Ainsi, la plage de fonctionnement où l'estimateur neuronal produit la plus grande erreur est paradoxalement celle où cette erreur a le moins d'impact sur la prise de décision.

4.3.6.1.5 Remarques sur la précision de l'estimation

Précédemment, il a été fait mention de l'occurrence de deux phases lors de l'apprentissage : une phase de généralisation et une phase de spécialisation. Dans le cas d'une base de donnée de petite taille, la phase de spécialisation peut se transformer en phase de sur-apprentissage lorsque la phase d'apprentissage est prolongée excessivement pour améliorer l'estimation du réseau neuronal.

Ce problème de sur-apprentissage est surmonté dès lors que la taille de la base de données augmente. En effet, un sur-apprentissage sur une base de données exhaustive de tous les points de fonctionnement du système n'est rien de plus que l'objectif primaire visé lors de la conception de l'estimateur.

4.3.6.2 Rapidité de l'estimation

Le tableau 4.2 dresse un bilan des performances de l'estimateur en terme de précision et de rapidité. Le speedup lié à l'utilisation de l'estimateur est plus important lorsque le nombre de points de fonctionnement à estimer est faible.

Par la suite, nous privilégierons l'emploi de l'estimateur sur de faibles horizons temporels d'une part pour maximiser le speedup et d'autre part car il n'est pas pertinent d'effectuer des prédictions sur un horizon trop lointain du fait de l'occurrence de perturbations de trafic.

	Durée de résolution itérative	Durée d'estimation par approximateur neuronal	Speedup	MAE finale
Cas 1 ($N_{data} \approx 3.10^3$)	18,23 s	0,14 s	130,2	0,0178
Cas 2 ($N_{data} \approx 4.10^4$)	205,80 s	2,76 s	74,6	0,0057

Tableau 4.2 – Récapitulatif des performances de l'estimateur.

Dans le deuxième cas d'étude, la MAE est plus faible que dans le premier cas d'étude, ce qui suggère qu'en moyenne l'estimation est plus précise lorsque la taille de

la base de données est plus importante.

Enfin, ce tableau valide la possibilité d'utiliser cette méthodologie pour une application où le temps de réponse souhaité est de l'ordre de la dizaine de secondes.

4.4 Optimisation dynamique des temps d'arrêts en station

4.4.1 Rappels des objectifs

Cette section a pour objectif d'adapter les travaux présentés au chapitre 3 sur l'optimisation offline des temps d'arrêt en station pour les rendre utilisables dans une application en temps réel.

Pour cela deux défis doivent être relevés. D'une part réduire le temps de calcul des méthodes d'optimisation et d'autre part intégrer les aléas d'exploitation qui peuvent survenir dans le processus décisionnel.

Deux options peuvent être envisagées : premièrement, une amélioration des caractéristiques de l'unité de calcul et deuxièmement un apprentissage des résultats issus de l'optimisation offline. Dans un contexte industriel, cette première solution serait envisageable mais non souhaitable pour des raisons de logistique et de coûts.

La deuxième solution est assez semblable à celle qui a été mise en œuvre dans la section précédente et consiste à déduire des résultats de l'optimisation une politique de décision robuste permettant de définir le temps de stationnement optimal de chaque train en station.

Généralement, les travaux portant sur la commande optimale adoptent une démarche qui consiste à calculer une trajectoire optimale puis à construire un contrôleur pour suivre la consigne du système. Cette approche est adaptée à certains domaines comme l'aéronautique ou l'animation 3D, où une trajectoire optimale peut être connue a-priori.

Cependant, dans de très nombreux domaines comme la robotique ou la finance, la dynamique du système n'est généralement pas prévisible, ce qui rend cette approche inefficace.

Pour palier à ces inconvénients, de nouvelles méthodes mathématiques et algorithmiques ont vu le jour pour calculer en ligne de nouvelles trajectoires optimales et ainsi adapter la stratégie suivie par le système.

[172] dresse ainsi une étude assez complète concernant les méthodes mathématiques qu'il est possible d'utiliser pour réaliser cet objectif.

4.4.2 Etat de l'art sur l'optimisation dynamique

Dans des conditions réelles d'exploitation, des perturbations de trafic influencent les temps de parcours des trains, ainsi que les horaires d'arrivées et de départs en stations. Une régulation de trafic est utilisée pour gérer ces perturbations de trafic et permettre aux trains de respecter au maximum la table horaire initiale. Ainsi, un aléa subi par un train peut être propagé à d'autres trains présents sur le réseau pour respecter les

contraintes techniques d'exploitation définies par l'exploitant.

Dans cette section, une table horaire est dite robuste si celle-ci a la capacité d'absorber les aléas de trafic qui se produisent en temps réel.

Un grand nombre d'études ont été dédiées à la conception de tables horaires robustes comme [173], [174],[175] ou encore [176]. Cependant, aucune planification hors-ligne ne peut être assez robuste pour absorber tous les types d'aléas sans compromettre les performances de la table horaire. De fait, il est nécessaire de modifier en temps réel le planning horaire initial pour s'adapter aux modifications imposées par les perturbations de trafic. Néanmoins, dans la pratique, la robustesse d'une table horaire est largement dépendante de la régulation de trafic utilisée.

En 2015, Corman [177] a établi un état de l'art très exhaustif des travaux concernant les différentes approches employées dans la littérature pour effectuer une replanification en temps réel des tables horaires de lignes ferroviaires. De cette étude, plusieurs enseignements peuvent être tirés.

D'une part, il est nécessaire d'avoir une bonne connaissance en temps réel du système ferroviaire à replanifier (position, profil de vitesse, aléas de trafic,...). Ce premier point est un pré-requis indispensable pour envisager une replanification efficace.

D'autre part, une vision probabiliste des états futurs du système est obligatoire pour obtenir une aide à la décision pertinente et précise. En effet, sans connaissance de l'évolution future du réseau ferroviaire, l'optimalité de la replanification ne peut être assurée et est même fortement compromise, puisque cela revient à effectuer une étude statique d'un système dynamique en connaissant uniquement les informations du système au pas de temps courant. Cette vision probabiliste est assimilable à la capacité d'anticiper l'évolution du système pour effectuer la replanification.

En outre, [177] met en lumière que la nature de la modélisation influence également les performances de la replanification, que ce soit pour l'atteinte des objectifs initiaux, la rapidité de la prise de décision ou le degré de précision de la planification (optimalité de la prise de décision sur un horizon temporel).

Enfin, cette étude démontre surtout qu'il existe une très grande quantité d'approches différentes pour réaliser une même macro-tâche. Chacun de ces travaux présente des spécificités, des avantages et désavantages qui dépendent avant tout des informations disponibles pour effectuer une replanification optimale en temps réel ainsi que des objectifs visés. De fait, aucune méthode universelle ne résulte de cet état de l'art, pour effectuer une optimisation dynamique des temps d'arrêt en station dans une ligne ferroviaire de type métro automatique.

L'optimisation temps réel des temps d'arrêt en station semble donc être liée à deux contraintes majeures : la rapidité d'exécution de la boucle d'optimisation et la connaissance probabiliste de l'évolution future de l'état du système.

Il a alors été choisi d'étudier la possibilité d'utiliser une approche par apprentissage par renforcement (AR) pour réaliser cette tâche d'optimisation en temps réel.

4.4.3 Définition de l'apprentissage par renforcement

De tous temps, l'homme a utilisé un système de récompense/punition (également appelés *renforceurs*) pour conditionner les actions des animaux qu'il souhaitait domestiquer.

En associant ces signaux de renforcement aux actions de l'animal, il est possible de lui faire adopter des comportements différents suivant la nature du renforceur : l'animal va chercher à maximiser les récompenses reçues et à minimiser ses punitions [178].

Le cerveau humain exploite ce principe de récompense/punition à travers la sécrétion de dopamine produite par les ganglions de la base afin de renforcer les connections synaptiques entre les neurones. Les travaux de Berridge et Robinson [179] constituent en ce sens une étude intéressante sur le fonctionnement neuro-chimique du système nerveux humain.

L'apprentissage par renforcement définit l'interaction entre un agent et son environnement : dans un état s_n de l'environnement, l'agent choisit et exécute une action a_n ce qui provoque une transition vers l'état s_{n+1}. L'agent reçoit alors le signal de renforcement r_n correspondant au bénéfice que l'action a_n a eu sur l'atteinte de l'objectif de l'agent.

Ce signal de renforcement est ensuite utilisé pour améliorer la politique de décision, à savoir, la séquence d'action optimale à effectuer pour maximiser les récompenses reçues et ainsi atteindre l'objectif fixé.

Dans cette section, un épisode est défini comme une série d'expériences permettant à un agent de partir d'un état initial et d'arriver à un état terminal. L'agent se déplace d'état en état en effectuant des actions et en observant les signaux de renforcement qui vont orienter sa recherche d'une politique optimale.

4.4.3.1 Processus de décision markovien

En 1957, Bellman pose les bases de l'apprentissage par renforcement en introduisant la notion de programmation dynamique, dans le but de concevoir des méthodes de contrôle optimal pour des systèmes dynamiques discrets et stochastiques. Dans [180], il explicite la notion de *processus de décision markovien* (PDM), afin de caractériser l'évolution d'un système suivant une succession d'états distincts, où les actions entreprises dépendent d'une fonction probabiliste de transition.

La figure 4.17 illustre le déroulement d'un PDM pour un agent qui part d'un état initial s_0 pour atteindre un état final s_n. Chacune des actions entreprises dans les différents états amène l'agent à recevoir une récompense.

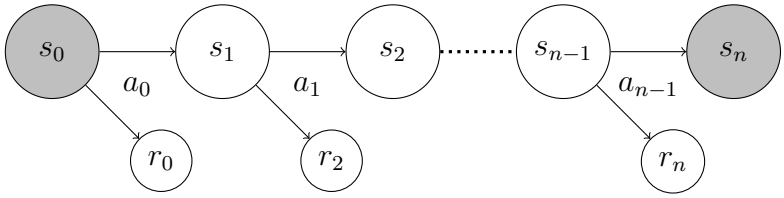


FIGURE 4.17 – Illustration d'un processus de décision markovien.

En 1988, Sutton et Barto effectuent une formalisation mathématique des algorithmes d'apprentissage par renforcement et généralisent le principe de commande optimale par l'utilisation de PDM [181], [182].

Un PDM se caractérise par le fait que l'agent doit résoudre séquentiellement des problèmes de décisions, où chaque décision courante influence la résolution des problèmes suivants et également, par l'incertitude des conséquences de chacune des actions possibles dans un état donné.

Ce type de problème est formalisé mathématiquement par le quadruplet $\{S, A, P, R\}$, où S est l'ensemble des états du système, A est l'ensemble des actions pouvant être accomplies par l'agent, P est une fonction de transition représentant la probabilité pour l'agent de se retrouver dans l'état suivant s' après avoir effectué l'action a dans un état s et R est la fonction de récompense du système représentant la probabilité pour l'agent de recevoir une récompense r en ayant effectué l'action a dans un état s. Une variable T représentant l'axe temporel des décisions peut également être introduite lorsque l'évolution du système ou les paramètres du PDM sont dépendants du temps [183].

Un problème dépendant du temps se caractérise notamment par le principe de causalité : il n'est pas possible de revenir en arrière une fois qu'une action a été effectuée.

La résolution d'un PDM consiste alors à contrôler l'agent pour qu'il effectue les actions lui permettant d'optimiser les récompenses reçues tout au long de son interaction avec l'environnement.

La notion de *politique* ou *stratégie*, notée π, est introduite pour qualifier la décision qui amène l'agent à effectuer une action dans un état donné. Un état du système est alors défini par (4.24), où $\pi(s_t)$ représente l'action effectuée dans l'état s_t sous la politique π.

$$s_{t+1} = P(s_t, \pi(s_t)) = P(s_t, a_t) \tag{4.24}$$

4.4.3.2 Critères de performance

La définition de critères de performance ambitionne de caractériser la politique suivie par l'agent apprenant. Différents critères de performances peuvent être employés pour évaluer une politique en mesurant le cumul des signaux de renforcement espérés le long d'une trajectoire.

Un critère fini C_f est une somme des récompenses immédiates et futures qui assigne le même poids à toutes les étapes de la trajectoire tandis qu'un critère actualisé C_a introduit la notion de facteur de dépréciation pour régler l'importance des récompenses futures par rapport aux récompenses immédiates.

L'expression de l'espérance du critère fini est rappelée par l'équation (4.25) où m est la longueur de la trajectoire à optimiser. L'expression du critère actualisé est donnée par l'équation 4.26.

$$C_f = E[\sum_{n=0}^{m} r_n | \pi, s_0] \tag{4.25}$$

$$C_a = E[\sum_{n=0}^{\inf} \gamma^n r_n | \pi, s_0] \tag{4.26}$$

Le facteur γ simule la *confiance* que l'on peut avoir dans l'estimation d'une récompense future probable ; plus la valeur de γ est élevée, plus l'agent aura confiance dans la vision à long terme des récompenses reçues en suivant la trajectoire définie par la politique π.

Le critère actualisé est particulièrement utilisé pour traiter les PDM où il existe une forte incertitude sur les décisions prises sur un horizon lointain en suivant la politique courante. Dans notre étude, l'utilisation de ce critère permet de probabiliser la vision à long terme et est donc privilégiée par rapport au critère fini.

D'autres critères comme le critère total (cumul des récompenses obtenues sur un temps infiniment long) et le critère moyen (moyenne des récompenses sur l'horizon de prédiction) peuvent également être employés mais ne sont pas développés ici, car ils ne présentent pas d'intérêt pour l'étude.

4.4.3.3 Fonction valeur

La *fonction valeur* en un état donné correspond à l'espérance des récompenses immédiates et futures en suivant la politique π. Pour un critère actualisé, la fonction valeur est définie comme la somme pondérée des récompenses futures en suivant la politique π à partir d'un état s_0 (4.27), à savoir l'espérance du critère actualisé.

$$V^{\pi}(s_0) = \sum_{n=0}^{\inf} \gamma^n R(s_n, a_n) \tag{4.27}$$

Le problème de contrôle optimal consiste alors à déterminer la politique optimale notée π^* qui maximise les récompenses perçues par l'agent. Pour cela, il s'agit de sélectionner les actions qui maximisent les récompenses reçues. La fonction valeur vérifie alors l'équation de Bellman (4.28), dont l'unique solution est la fonction valeur optimale que l'on note $V^{\pi*}$.

Le principe d'optimalité de Bellman appliqué à la fonction valeur permet de relier la valeur d'un état à la valeur de tous les états consécutifs qui peuvent être visités.

$$V^{\pi^*}(s) = \max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi^*}(s') \quad \forall s \in S \tag{4.28}$$

De (4.28), il est possible de déduire la définition de la politique optimale π^* (4.29).

$$\pi^*(s) = \arg\max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi^*}(s') \quad \forall s \in S \tag{4.29}$$

4.4.3.4 Fonction de valeur état-action

Un agent peut également caractériser ses interactions avec l'environnement via la *fonction de valeur état-action* Q, que l'on peut retrouver sous l'appellation *fonction Qualité*. Cette fonction mesure la qualité d'une paire état-action pour une politique π fixée.

La fonction valeur se définit ainsi comme la moyenne des $Q^\pi(s,a)$ pondérées par la probabilité de chaque action (4.30a).

La fonction Q est particulièrement utile pour déterminer le comportement optimal le plus probable d'un agent situé dans un état courant. L'expression de la fonction état-action optimale est reprise par (4.30b).

$$\begin{cases} V^\pi(s) = \sum_a \pi(s,a) Q^\pi(s,a) & \text{(4.30a)} \\ Q^{\pi^*}(s,a) = (R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi^*}(s') & \text{(4.30b)} \end{cases}$$

4.4.4 Paramétrage de l'apprentissage par renforcement

4.4.4.1 Model-free vs Model-based / exploration vs exploitation

Il convient de différencier les méthodes d'apprentissage par renforcement utilisant un modèle de l'environnement (*model-based*) et celle sans modèle (*model-free*). L'apprentissage model-based a pour objectif d'apprendre les fonctions de transitions P et de récompenses R pour déterminer la stratégie optimale π^*. A l'inverse, l'apprentissage model-free tente d'estimer directement la fonction de valeur état-action Q^* sans étudier les fonctions P et R.

Ces deux modes d'apprentissage par renforcement s'opposent par quelques propriétés caractéristiques. L'apprentissage model-based nécessite une longue phase d'exploration pour déterminer précisément pour chaque point de fonctionnement du système les fonctions P et R afin d'en déduire ensuite une politique décisionnelle optimale. Dans le cas de problèmes possédant un vaste espace état-action, ce type d'apprentissage peut donc s'avérer extrêmement gourmand en temps de calcul voire, dans le pire des cas, impossible à mettre en œuvre pour explorer l'ensemble des couples état-action.

L'apprentissage model-free, est une approche gloutonne dans le sens où lors de l'apprentissage, l'agent va tenter de maximiser les récompenses reçues en exploitant systématiquement les actions menant aux valeurs de $Q(s,a)$ les plus grandes. Cette méthode est donc à dominante exploitante et est en théorie plus rapide qu'une méthode model-based, puisqu'il n'est pas nécessaire d'explorer tous les couples état-action possibles pour obtenir une politique décisionnelle optimale [8] .

En pratique, il n'est pas souhaitable d'utiliser une méthode purement exploratrice ou exploitante. En effet, si on se ramène au problème d'optimisation de la consommation énergétique d'une ligne de métro, il peut s'avérer nécessaire de dégrader une synchronisation de trains à un instant pour générer une meilleure synchronisation dans le futur.

En outre, en considérant un exemple totalement décorrelé de la présente étude : l'apprentissage d'un jeu d'échec, il est parfois nécessaire de sacrifier une pièce pour se trouver par la suite dans une position plus favorable.

8. Plus généralement, les politiques obtenues par cette approche sont dites sub-optimales car pour des problèmes complexes, il est possible de juger si une politique mène à l'objectif fixé, mais il n'est souvent pas possible de connaitre la politique optimale du problème.

Cette notion est connue sous le nom de compromis exploration-exploitation. L'étude de ce compromis a fait l'objet de nombreuses publications comme [184].

Lorsque l'atteinte de l'objectif nécessite de passer par de nombreux états présentant des récompenses négatives ou de faibles signaux de renforcement, une politique à dominante exploratrice est plus appropriée. A l'inverse, dans le cas de jeu de plateaux (échecs, othello, backgammon,...), une stratégie basée majoritairement sur l'exploitation peut s'avérer plus bénéfique pour trouver la politique optimale.

Dans un problème industriel présentant des fortes non linéarités, une hybride entre exploration et exploitation semble être l'approche la plus appropriée qui, bien que nécessitant un temps de calcul plus important qu'une stratégie gloutonne, permet d'obtenir une meilleure connaissance du système étudié.

Dans la suite du manuscrit, une fonction de transition basée sur une distribution de Boltzmann-Gibbs est utilisée (4.31) [185], [186].

$$P(s,a) = \frac{e^{Q(s,a)/\tau}}{\sum\limits_{b \in A(s)} e^{Q(s,b)/\tau}} \tag{4.31}$$

Dans l'équation (4.31), le terme τ est une variable dont le but est de moduler le caractère stochastique de la fonction de transition P. Une valeur élevée de τ entraîne une sélection aléatoire tandis qu'une valeur faible tend à rendre la sélection des actions gloutonne. De plus, de par sa définition, cette règle de sélection présente l'avantage de donner plus de place à l'exploration des actions possibles quand les valeurs de $Q(s,a)$ sont dans le même ordre de grandeur, et de privilégier l'exploitation quand une action semble être beaucoup plus bénéfique que les autres. De plus, l'utilisation d'une règle de décroissance progressive du facteur τ permet d'accélérer la convergence de certaines méthodes, introduites dans la suite du développement, vers la politique optimale en favorisant aux premières itérations l'exploration puis l'exploitation des résultats dans les dernières itérations [187].

4.4.4.2 Caractéristiques de l'environnement

Suivant le problème traité, l'interaction entre l'agent et son environnement peut être caractérisée selon différents aspects :

Le niveau de perception de l'agent. Dans certains cas de figures, il peut être difficile pour l'agent de percevoir l'intégralité de l'état de l'environnement.

Le déterminisme de l'espace état-action. L'environnement est dit déterministe lorsque l'état suivant du système est complètement défini par le couple état-action courant. Cependant, même dans cette éventualité, la perception qu'a l'agent de son environnement peut être incomplète s'il n'a pas accès à l'intégralité des informations concernant l'état courant de l'environnement.

La dynamique du système étudié. Un environnement est dit dynamique lorsque l'état du système peut varier au cours de la prise de décision et il est dit statique dans le cas contraire.

La nature de l'interaction. L'interaction agent-environnement est dite épisodique lorsque celle-ci peut être divisée en événements appelés épisodes. Un épisode consiste alors en un cycle perception-action-renforcement : l'agent perçoit l'état

du système puis choisit l'action appropriée à effectuer et reçoit enfin une récompense de son environnement.

La nature de l'espace état-action. Lorsque le nombre d'état et d'action possibles est fini, cet espace est dit discret, et il est continu dans le cas inverse.

L'analyse de ces aspects permet de sélectionner la méthode ou l'algorithme le plus adapté pour solutionner le problème étudié.

4.4.5 Programmation dynamique

La programmation dynamique est une méthode algorithmique dont la philosophie est de considérer qu'un problème d'optimisation est composé d'un ensemble de sous-problèmes. La solution optimale du problème global est alors obtenu à partir des solutions optimales des sous-problèmes.

Sous l'hypothèse d'un PDM avec des ensembles S et A finis, les équations issues du principe d'optimalité de Bellman permettent d'estimer les fonctions V^{π^*} et Q^{π^*} afin d'en déduire la politique optimale π^*

Dans le cadre de la résolution d'un PDM par programmation dynamique, il existe deux approches : l'itération sur les valeurs et l'itération sur les politiques.

L'itération sur les valeurs est l'approche la plus classique et se base sur l'utilisation de la formulation du principe d'optimalité de Bellman appliqué à la fonction valeur (4.32). Une légère modification de la notation de l'équation (4.28) permet de mettre en évidence une suite récurrente qui converge vers V^{π^*} [188].

$$V^{\pi^*}{}_{n+1}(s) = \max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi^*}{}_{n}(s') \tag{4.32}$$

Le principe de l'itération sur les politiques est d'utiliser une politique π_n, puis d'évaluer la fonction valeur V^{π_n} issue de cette politique et d'améliorer cette politique par une règle gloutonne pour déterminer la politique suivante π_{n+1} (4.33). L'itération se termine lorsqu'aucune évolution n'est observée entre deux politiques successives.

$$\pi_{n+1}(s) = \arg\max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_n}(s') \tag{4.33}$$

Le cadre de la programmation dynamique correspond à la résolution la plus simple et intuitive d'un PDM. D'autres méthodes dérivées de la programmation dynamique ont également été introduites par la suite pour solutionner des cas plus complexes où la dynamique du système n'est pas connue ou difficilement identifiable [188].

4.4.6 Méthodes de Monte-Carlo

Une méthode de Monte-Carlo (MC) est une technique algorithmique destinée à fournir une valeur approchée d'une quantité déterministe via un procédé probabiliste.

Un paradigme largement répandu pour illustrer cette méthode est l'estimation des probabilités de chaque issue du jeu *Pile ou Face*. En lançant un grand nombre de fois une pièce équilibrée, cela devrait mener à une équiprobabilité de tomber sur le côté pile ou le côté face de la pièce. De fait, en réalisant suffisamment d'essais d'une expérience,

il serait possible d'obtenir une estimation du comportement moyen du système lors de cette expérience.

La méthode MC consiste alors à réaliser un grand nombre d'expériences et à enregistrer pour chacune d'entre elles, les transitions et actions effectuées. Ainsi, au fur et à mesure des expériences, la fonction valeur associée à chaque état se rapproche de la valeur exacte de l'état pour la politique suivie. L'estimation de la fonction V s'apparente ainsi à un moyennage des récompenses reçues lors des différentes expériences.

En utilisant un critère actualisé, la fonction récompense R pour chaque épisode est définie par (4.34a) et la fonction valeur V est mise jour par la règle (4.34b), où m est la longueur de la trajectoire entre l'état courant et l'état terminal, tandis que M est le nombre d'expériences effectuées.

$$\begin{cases} R^n = \sum_{k=0}^{m} \gamma^k r_k & \text{(4.34a)} \\ V^\pi(s) = \frac{1}{n} \sum_{n=1}^{M} R^n & \text{(4.34b)} \end{cases}$$

Comme le rappelle [189], une méthode de Monte-Carlo est généralement utilisée pour calculer les propriétés statiques d'un système et non ses propriétés dynamiques puisqu'il est nécessaire de réaliser un certain nombre d'expériences pour en déduire une politique efficace. Cependant, sous l'hypothèse d'un nombre d'expériences suffisant, une méthode MC peut s'avérer efficace pour étudier la dynamique d'un système.

4.4.7 Méthodes de différences temporelles

Les méthodes de différences temporelles (TD pour *Temporal Difference*) sont une hybridation des méthodes de programmation dynamique et de Monte-Carlo, présentées initialement dans les travaux de [190]. Une méthode TD définit ainsi une politique décisionnelle en se basant sur une série d'expériences et sur les observations du système pour différents points de fonctionnement.

L'apprentissage par TD se subdivise en deux catégories : la prédiction et le contrôle. Le problème de prédiction consiste à déterminer la récompense actualisée en suivant la politique π à partir d'un état s, ce qui revient à prédire la fonction valeur V. Le problème de contrôle, quant à lui, consiste à apprendre la récompense actualisée obtenue en suivant la politique π, en effectuant l'action a dans un état s, ce qui revient à déterminer la valeur de la fonction Q.

La variante TD(0) est la forme la plus simple de la méthode TD. L'argument 0 de la méthode désigne le fait que l'agent a une vision minimaliste : en suivant la politique π dans un état s_k, l'agent n'a accès qu'aux informations r_k et s_{k+1}. Ces informations permettent ensuite de mettre à jour l'approximation de la fonction V, en évaluant l'évolution de V en passant de s_k à s_{k+1} via le système (4.35), où α est le taux d'apprentissage.

$$\begin{cases} {\epsilon^V}_{prediction} = r_{t+1} + \gamma {V^\pi}_n(s_{k+1}) - {V^\pi}_n(s_k) & \text{(4.35a)} \\ {V^\pi}_{n+1}(s_k) = {V^\pi}_n(s_k) + \alpha\, {\epsilon^V}_{prediction} & \text{(4.35b)} \end{cases}$$

Par rapport à la méthode de Monte-Carlo qui attend la fin d'un épisode pour mettre à jour la fonction V, cette méthode de mise à jour permet de recalculer une nouvelle politique au fil des observations de l'agent et ainsi orienter la recherche de l'agent vers les états jugés les plus prometteurs.

Néanmoins, dans un environnement dont l'évolution est incertaine, il est nécessaire de connaitre la probabilité de transition. Ainsi, il s'avère plus pertinent d'effectuer un apprentissage de la fonction Q afin d'effectuer un contrôle plus efficace du problème pour que l'agent atteigne son objectif.

4.4.7.1 Mise à jour de la stratégie

L'utilisation de la fonction valeur état-action est privilégiée car elle permet un meilleur contrôle du problème en spécifiant clairement l'intérêt d'une action dans un état donné. La mise en équation de la fonction Q est très similaire à celle de la fonction V (4.36).

$$\begin{cases} \epsilon^{Q}{}_{prediction} = r_{t+1} + \gamma Q^{\pi}{}_{n}(s_{k+1}, a_{k+1}) - Q^{\pi}{}_{n}(s_k, a_k) & \text{(4.36a)} \\ Q^{\pi}{}_{n+1}(s_k, a_k) = Q^{\pi}{}_{n}(s_k, a_k) + \alpha\, \epsilon^{Q}{}_{prediction} & \text{(4.36b)} \end{cases}$$

L'évaluation et l'amélioration des stratégies déterminées par une méthode TD peuvent se faire de deux manières.

Une méthode *on-policy* La politique suivie est mise à jour au cours des expériences réalisées. L'estimation des performances de la politique se fait en prenant des décisions basées sur cette même politique, autrement dit, le coût de l'exploration est pris en compte lors de la mise à jour de la fonction Q.

Une méthode *off-policy* La stratégie optimale π^* est estimée en suivant une politique π gloutonne effectuant à chaque fois l'action qui maximise le signal de renforcement reçu.

Le choix de l'une ou l'autre de ces méthodes pour réaliser l'apprentissage d'une tâche dépend principalement de la nature du problème à résoudre. Dans un exemple présenté dans [190] mettant en compétition une méthode on-policy et une méthode off-policy sur un même problème avec des complexités différentes, les auteurs mettent en évidence la capacité d'une méthode on-policy à apprendre à éviter les mauvaises actions. Le problème considéré consiste à déterminer la trajectoire la plus rapide allant d'un point initial à un point final dans un environnement présentant des positions néfastes pour l'agent.

Un résumé des observations de [190] est proposé dans le tableau 4.3.

En outre, dans [191], l'auteur effectue une comparaison similaire sur un jeu d'othello et montre qu'une méthode on-policy est plus rapide qu'une méthode off-policy à converger vers une bonne politique bien que la courbe d'apprentissage de la première méthode soit assez instable.

La méthode off-policy souffre ainsi de l'exploitation d'une politique non optimale lors des essais effectués tandis qu'une méthode on-policy intègre les mauvaises décisions prises dans la mise à jour de la politique.

Le choix de l'une ou l'autre de ces méthodes reste donc assez délicat à effectuer puisque ce choix dépend du problème traité. Cependant, une méthode on-policy semble plus propice à traiter des problèmes de grande dimension.

Dimension du problème	On-policy	Off-policy
n = 10	• trajectoire optimale non déterminée • évite d'effectuer les actions menant à des états néfastes	• trajectoire optimale
n = 10000	• apprentissage d'une bonne trajectoire assez rapidement • évite d'effectuer les actions menant à des états néfastes	• trajectoire optimale très lente à déterminer

TABLEAU 4.3 – Comparaison des caractéristiques des méthodes on et off-policy pour deux tailles de problèmes

4.4.7.2 Une méthode off-policy : Q-learning

La méthode Q-learning est assez similaire à la méthode TD(0), à la différence que le Q-learning met à jour la fonction Q au lieu de la fonction V.

$$\begin{cases} \delta_{off} = r_k + \gamma \max\limits_{a_k \in A(s_k)} Q^{\pi}(s_{k+1}, a_{k+1}) - Q^{\pi}(s_k, a_k) \\ Q^{\pi}(s_k, a_k) = Q^{\pi}(s_k, a_k) + \alpha\, \delta_{off} \end{cases} \tag{4.37}$$

La règle de mise à jour de Q est donnée par (4.37). Il s'agit de la version la plus simple de Q-learning avec un seul pas de vue vers l'avant. Les actions de l'agent sont déterminées selon une politique π mise à jour par une règle gloutonne.

Cette méthode présente l'avantage de ne pas prendre en compte le coût d'exploration dans l'évaluation de la fonction Q, ce qui permet de constamment utiliser la meilleure politique déterminée par la méthode.

En outre, ici, une stratégie gloutonne de sélection des actions est utilisée, cependant comme il est expliqué en section 4.4.4.1, il est possible et même préférable d'utiliser une fonction de transition différente, notamment pour intégrer la notion d'exploration (4.31).

4.4.7.3 Une méthode on-policy : SARSA

À l'inverse de la méthode Q-learning, la méthode SARSA permet d'améliorer la politique décisionnelle au fil de l'eau. L'agent sélectionne l'action à entreprendre en suivant la politique courante π puis met à jour la valeur de la fonction Q. Cette méthode tire son nom du fait qu'elle nécessite de connaitre le quintuplet $\{s_k, a_k, r_k, s_{k+1}, a_{k+1}\}$ à chacune des étapes k de l'épisode.

$$\begin{cases} \delta_{on} = r_k + \gamma Q^{\pi}(s_{k+1}, a_{k+1}) - Q^{\pi}(s_k, a_k) \\ Q^{\pi}(s_k, a_k) = Q^{\pi}(s_k, a_k) + \alpha\, \delta_{on} \end{cases} \tag{4.38}$$

La règle de mise à jour de la méthode SARSA est définie par (4.38). Cette règle se démarque de celle du Q-learning par plusieurs caractéristiques. Premièrement, l'agent connait l'action suivante qui sera effectuée et prend en compte cette information pour la mise à jour de la fonction Q. Deuxièmement, la politique suivie π bénéficie d'une amélioration continue ce qui a pour effet d'accélérer l'apprentissage et la convergence vers la politique optimale du problème [185].

Une fonction de transition similaire à (4.31) peut également être utilisée pour inciter l'agent à effectuer une exploration, puis une exploitation dans les dernières étapes du processus itératif.

4.4.7.4 Algorithme type de TD-learning

L'algorithme 9 présente un algorithme généraliste pour la mise en oeuvre des deux variantes de TD-learning : la méthode SARSA et la méthode Q-learning. Dans cet algorithme, les états et actions de l'itération suivante sont respectivement notés s' et a'.

Algorithme 9 Algorithme générique de TD-learning à un pas

```
Initialiser la politique π suivie par l'agent
Pour tout a ∈ A , s ∈ S Faire
    Initialiser la fonction Q(s, a)
Fin du Pour
Pour tout épisodes Faire
    Initialiser l'état de départ s.
    Si méthode on-policy Alors
        a ← π(s)
    Fin du Si
    Tant que L'état final n'est pas atteint Faire
        Si méthode on-policy Alors
            Recevoir la récompense r et observer l'état suivant s'
            a' ← π(s')
            Mettre à jour la fonction Q avec la règle (4.37)
            s ← s'
            a ← a'
        Sinon
            a ← π(s)
            Recevoir la récompense r et observer l'état suivant s'
            Mettre à jour la fonction Q avec la règle (4.38)
            s ← s'
        Fin du Si
    Fin du Tant que
Fin du Pour
```

Il est à noter qu'en théorie, la convergence vers une politique optimale est assurée lorsque chaque couple état-action a été visité un nombre infini de fois afin de connaitre

précisément la meilleure séquence d'actions à effectuer à partir d'un état initial pour atteindre l'objectif fixé le plus rapidement possible [192].

Dans (4.37) et (4.38), les termes δ_{off} et δ_{on} sont à l'origine de l'appellation *différence temporelle*. En effet, ces termes agissent comme un signal d'erreur qui guide l'apprentissage : si ce signal est positif, cela signifie que la récompense obtenue est supérieure à celle attendue et inversement si le signal est négatif.

4.4.7.5 Dimensionnement du signal de renforcement

La définition du signal de renforcement est une propriété fondamentale intrinsèque de toute méthode d'apprentissage par renforcement. Comme le nom de la méthode l'indique, l'agent construit des déductions logiques à partir des récompenses ou des malus reçus au cours de ses interactions avec l'environnement. Un signal de renforcement mal dimensionné entraîne un apprentissage lent, aléatoire et imprécis.

Dans notre cas d'étude, la fonction récompense R est définie comme le taux de réutilisation de l'énergie issue du freinage électrique (4.39), où $E_{freinage}$ est la quantité d'énergie électrique théoriquement récupérable et $E_{dissipee}$ est la quantité d'énergie électrique dissipée lors du freinage.

$$R = \frac{E_{freinage} - E_{dissipee}}{E_{freinage}} \tag{4.39}$$

Le principe retenu est que chaque fois qu'un train arrive en station, il utilise une partie de son temps de stationnement pour choisir l'action à effectuer (la modification de temps de stationnement dans la station). Cette modification implique une certaine synchronisation des phases de freinage/accélération entre le train concerné par l'action et les autres trains en ligne.

Les récompenses sont dimensionnées pour évaluer le taux d'énergie électrique issu du freinage récupératif par rapport à ce qui aurait théoriquement pu être renvoyé. Ainsi, à chaque phase de freinage des trains est, assignée une estimation de R dont la probabilité d'occurence décroît en fonction de son éloignement temporel par rapport à la prise de décision. La probabilité d'occurrence d'un événement correspond ici à la proportion de chance de voir se réaliser les événements prévus par rapport à toutes les éventualités possibles.

Cependant, cette probabilité d'occurrence dépend de nombreux facteurs humains et techniques, ce qui la rend difficilement identifiable. En pratique, c'est le paramètre de diffusion λ qui assure l'évolution de cette probabilité.

Le dimensionnement de la récompense doit tenir compte d'au moins quatre aspects :

- un aspect local : la prise de décision a un impact sur le taux de récupération du train concerné par l'action
- un aspect global : la prise de décision a un impact sur le taux de récupération des autres trains présents sur la ligne
- un aspect court terme concernant le taux de récupération du freinage sur la prochaine interstation
- un aspect long terme concernant le taux de récupération du freinage sur les interstations suivantes. L'horizon de prédiction peut alors être étendu indéfiniment

dans le temps en appliquant des coefficients de diffusion pour simuler l'incertitude sur l'occurence des événements prévus.

Ces différents aspects sont ensuite reliés par une combinaison linéaire afin de calculer la récompense reçue par chaque train suite à la réalisation d'une action.

La figure 4.18 représente la puissance théorique consommée par un train sur 3 interstations. A l'état initial, le train arrive en station, la récompense locale à court terme représente le taux de récupération du freinage électrique lors de la première interstation. La récompense locale à moyen terme concerne le taux de récupération durant la deuxième interstation tandis que la récompense locale à long terme concerne celui de la dernière interstation.

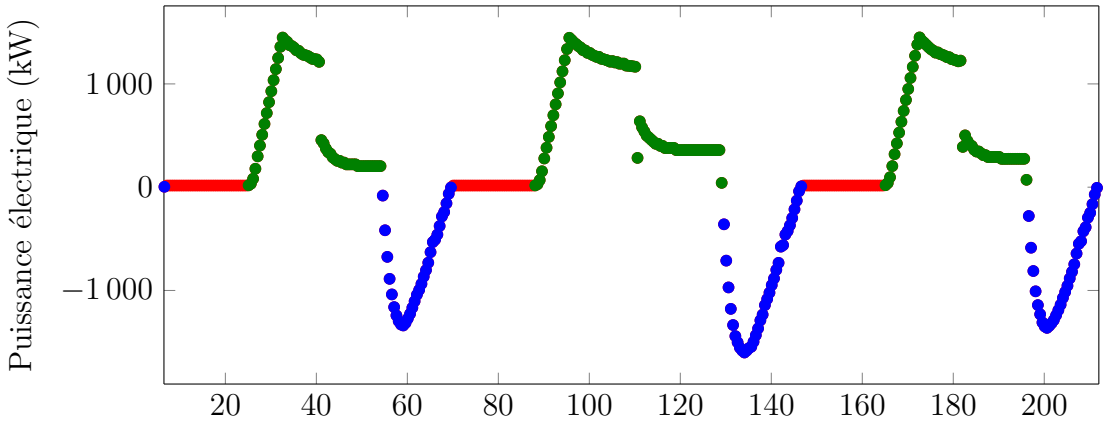


FIGURE 4.18 – Illustration du principe de récompense locale avec un horizon temporel de trois événements.

Les récompenses globales sont calculées sur les mêmes horizons temporels, mais cette fois en étudiant les taux de récupération de l'ensemble des trains du carrousel.

4.4.7.6 Exemple pratique

Le fonctionnement et la distribution des signaux de renforcement sont illustrés en considérant un cas d'étude. Un carrousel de 16 trains évoluant selon un intervalle donné est d'abord simulé, puis l'algorithme d'optimisation hybride est employé pour générer une distribution de combinaisons de temps de stationnement potentielles.

La solution optimale trouvée par l'algorithme est ensuite comparée à la solution nominale. La solution optimale permet un gain énergétique de 11% sur un tour de boucle, tandis que la combinaison nominale est la solution de référence utilisant la combinaison de temps d'arrêts nominaux (gain énergétique nul).

La figure 4.19 décrit l'évolution de la récompense locale à court terme $R^1{}_{t,local}$ vu par l'agent pour les deux combinaisons de temps d'arrêt en station sur 1000s de simulation.

L'analyse de l'évolution de la récompense locale court-terme indique que cette seule information n'est pas suffisante pour concevoir une politique efficace, en effet, l'impact énergétique de la solution optimale par rapport à la solution de référence n'est pas clairement visible.

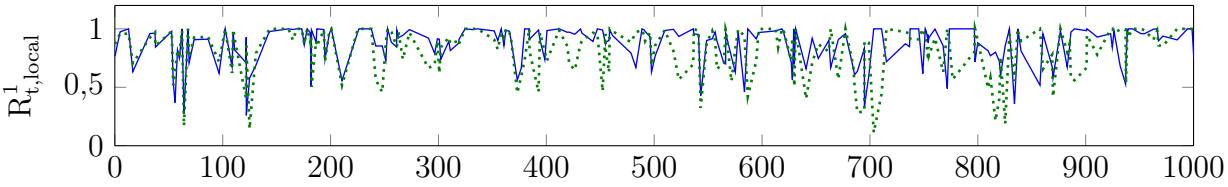
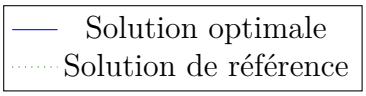


FIGURE 4.19 – Évolution des récompenses locales sur un horizon de 1000s d'exploitation.



La figure 4.20 représente l'évolution des récompenses globales pour trois horizons temporels : court ($R^1{}_{t,global}$), moyen ($R^2{}_{t,global}$) et long terme ($R^3{}_{t,global}$).

Cette figure permet de visualiser la supériorité de la solution optimale par rapport à la solution de référence pour réduire la dissipation de l'énergie du freinage.

Ces figures illustrent l'intérêt du partage des informations entre les trains. En effet, en considérant uniquement la composante locale du signal de renforcement (composante égoïste), la discrimination entre deux solutions extrêmes est difficilement réalisable.

En revanche, la composante globale (composante altruiste) fournit une information plus exploitable pour évaluer les impacts énergétiques de chacune des solutions.

En outre, comme le montre la figure 4.20, la solution optimale n'est pas dominante sur l'ensemble de l'horizon temporel.

Cela implique donc la nécessité de disposer d'un grand nombre de solutions potentielles pour en déduire la solution optimale du problème qui dominera toutes les autres quelles que soient les conditions d'exploitation.

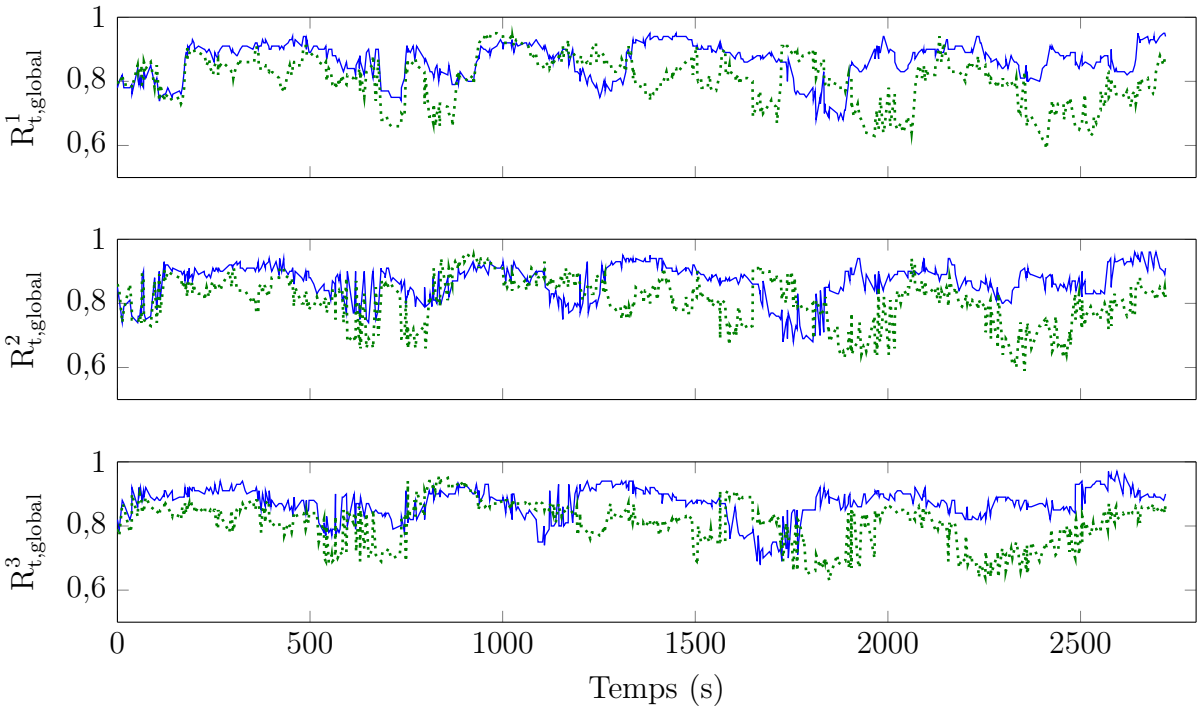


FIGURE 4.20 – Évolution des récompenses globales sur un tour de boucle complet.

4.4.8 Traces d'éligibilité

Dans les méthodes de TD-learning présentées précédemment, il a été supposé que le signal de renforcement reçu par l'agent ne concerne que la dernière transition d'état effectuée. Cependant, dans notre cas d'étude comme dans de nombreuses applications, l'état courant du système résulte d'un cheminement de l'agent parmi les états précédents et donc des actions effectuées lors des transitions précédentes. Il peut donc s'avérer utile de garder en mémoire les transitions effectuées pour caractériser la fonction Q à partir de la trajectoire globale empruntée par l'agent lors de l'apprentissage.

Les *traces d'éligibilité* exploitent ce concept en permettant de modifier la fonction Q pour l'ensemble des couples état-action visités par la trajectoire courante. De cette manière, la politique issue de l'apprentissage développe une aptitude d'anticipation et de prévision, tout en accélérant le processus d'apprentissage [192]. D'un point de vue théorique, les traces d'éligibilité font le lien entre les méthodes MC et TD.

L'un des points faibles des méthodes précédentes qui a été mis en lumière est la nécessité de réaliser un très grand nombre d'épisodes pour s'assurer d'avoir déterminé la politique optimale. Dans la méthode de Monte-Carlo, la mise à jour de la fonction V de la trajectoire suivie est effectuée une fois que l'état terminal a été atteint, ce qui permet lors de l'expérience suivante de bénéficier des connaissances du système acquises précédemment. De fait, en adaptant cette propriété aux méthodes TD, il devrait être intuitivement possible d'améliorer leurs propriétés de convergence.

4.4.8.1 Méthode TD(λ)

Un moyen de développer une politique qui possède une mémoire des états précédemment visités consiste à stocker les couples état-action visités et à propager la récompense courante aux autres transitions en les pondérant par un facteur de diffusion λ : l'erreur d'apprentissage est reportée en arrière dans le temps proportionnellement à l'ancienneté par rapport au couple état-action courant avec un amortissement régi par la valeur de λ.

La trace d'éligibilité $e_k(s,a)$ d'un couple état-action (s,a) indique à quel point la récompense courante influence la valeur de $Q(s,a)$ à la k-ième transition. Les états récemment visités doivent donc être plus fortement impactés par la récompense courante, de fait, la trace d'éligibilité est dégradée d'un facteur $\lambda\gamma$ à chaque transition.

Deux types de traces d'éligibilité sont couramment employés : les traces accumulatives et les traces avec réinitialisation.

4.4.8.2 Trace d'éligibilité accumulative

L'évolution de la trace d'éligibilité accumulative au cours des transitions est définie par le système (4.40). La trace d'éligibilité est incrémentée de 1 pour le couple état-action courant et une loi de décroissance est appliquée pour les autres couples précédemment visités.

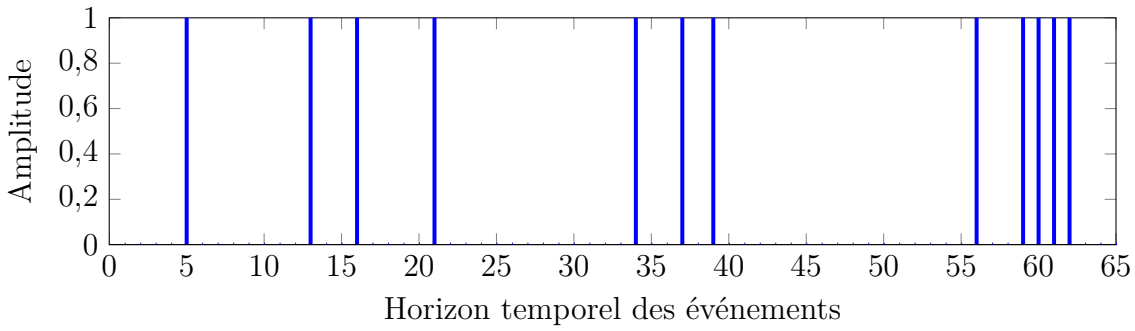
$$e_k(s,a) = \begin{cases} \lambda\gamma e_{k-1}(s,a) + 1 & \text{si } (s,a) = (s_k, a_k) \\ \lambda\gamma e_{k-1}(s,a) & \text{si } (s,a) \neq (s_k, a_k) \end{cases} \quad \forall s \in S, \forall a \in A(s) \tag{4.40}$$

Cependant, lorsqu'un couple état-action est visité plus d'une fois lors d'un épisode, la valeur correspondante de $e_t(s, a)$ devient largement plus grande que 1, ce qui a pour effet de laisser croire que l'action a effectuée dans l'état s est plus profitable qu'elle ne l'est en réalité en produisant une haute valeur de $Q(s, a)$ pour le couple considéré.

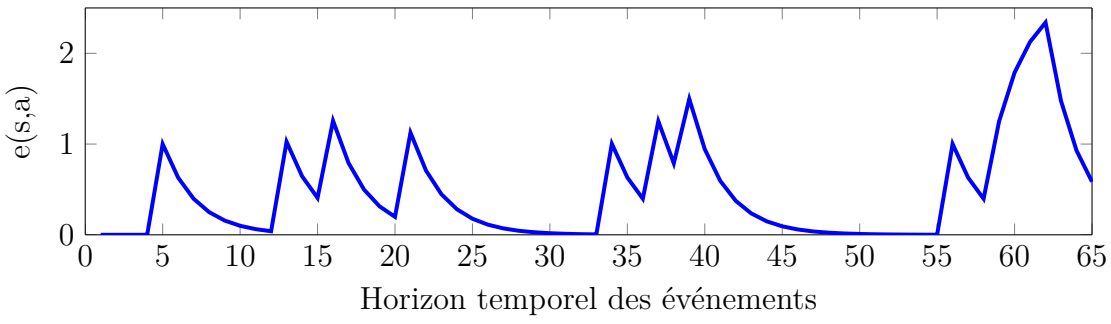
[193] a ainsi proposé une version améliorée de trace d'éligibilité : la trace d'éligibilité avec réinitialisation.

4.4.8.3 Trace d'éligibilité avec réinitialisation

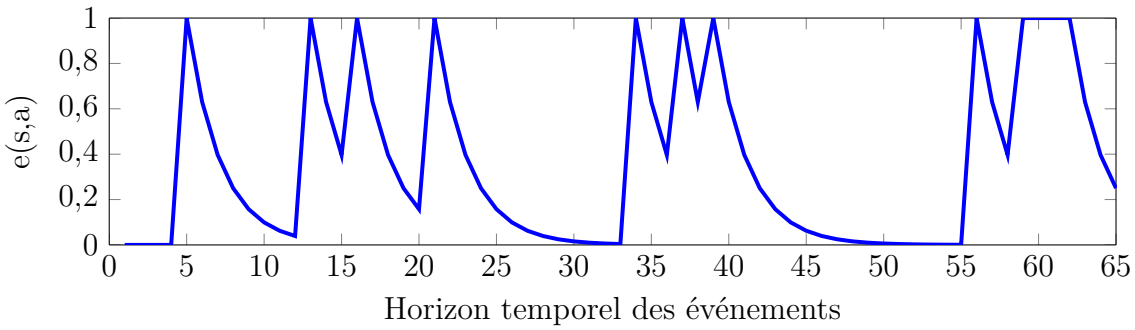
La trace d'éligibilité avec réinitialisation consiste à mettre à 1 la trace du couple courant au lieu de l'incrémenter pour éviter l'explosion de la valeur de $e(s, a)$ et de forcer à 0 la trace des actions non effectuées dans l'état s (4.41). La principale justification de cette variante fournie par [193] est que dans un environnement markovien, seule l'action effectuée dans l'état considéré est responsable des signaux de renforcement reçus par l'agent.



(a) Occurence des visites du couple (s,a) par l'agent



(b) Profil de la trace d'éligibilité accumulative



(c) Profil de la trace d'éligibilité avec réinitialisation

FIGURE 4.21 – Évolution des traces d'éligibilité en fonction de l'occurrence des visites du couple (s, a) par l'agent.

La figure 4.21 illustre l'évolution des deux variantes de traces d'éligibilité en fonc-

tion de la visite par l'agent du couple (s, a).

$$e_k(s,a) = \begin{cases} 1 & \text{si } (s,a) = (s_k, a_k) \\ 0 & \text{si } s = s_k \text{ et } a \neq a_k \\ \lambda\gamma e_{k-1}(s,a) & \text{si } s \neq s_k \end{cases} \quad \forall s \in S, \forall a \in A(s) \tag{4.41}$$

La règle de mise à jour de la fonction valeur état-action devient alors 4.42. À chaque étape k de l'épisode courant, cette règle doit être exécutée pour tous les couples état-action visités précédemment.

$${Q^\pi}_{k+1}(s,a) = {Q^\pi}_k(s,a) + \alpha\, \delta_{TD}\, e_k(s,a) \tag{4.42}$$

Le terme δ_{TD} fait référence indifféremment aux termes δ_{off} et δ_{on} définis par les systèmes (4.37) et (4.38).

Il existe dans la littérature quantité d'autres variantes des méthodes de TD-learning présentées précédemment intégrant ou non des traces d'éligibilité. Ces variantes ne sont pas explicitées car elles n'ont pas clairement démontré une efficacité[9] supérieure par rapport aux méthodes de bases énoncées dans ce manuscrit.

4.4.8.4 Récapitulatif des méthodes d'apprentissage par renforcement

La figure 4.22 présente un récapitulatif des méthodes d'apprentissage qui peuvent être envisagées pour résoudre des problèmes de type PDM, et de celles que nous avons privilégié. Il ne s'agit que d'un aperçu macroscopique et simplifié puisque dans la littérature il existe une grande quantité d'algorithmes de type TD-learning qui ont été formulés.

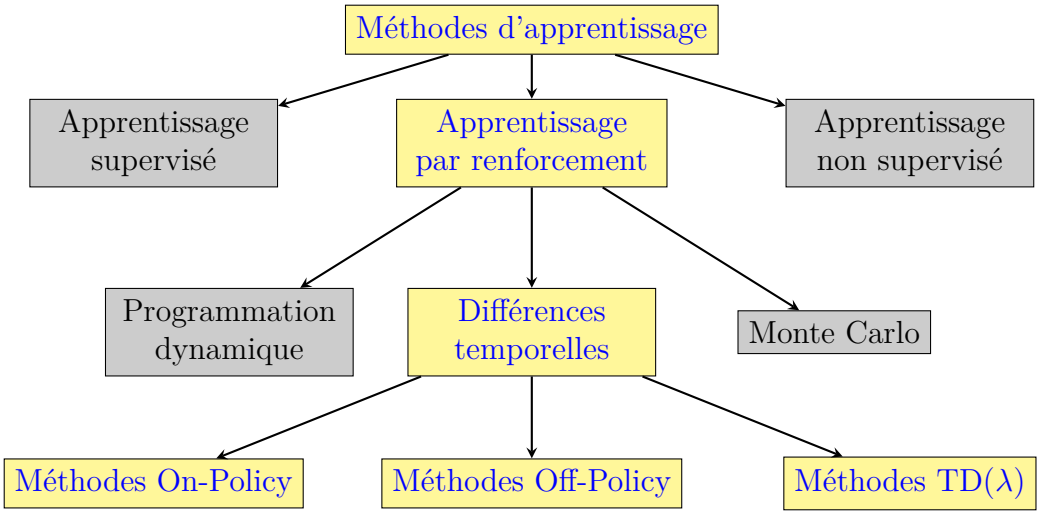


FIGURE 4.22 – Apercu des méthodes d'apprentissage par renforcement.

Par la suite cette arborescence pourra être poursuivie pour intégrer les méthodes utilisant des fonctions d'approximation pour stocker les informations inhérentes au problème traité.

9. en temps de calcul et en précision

4.5 Apprentissage par renforcement avec un réseau de neurones

Une méthode d'apprentissage par renforcement comme celle définie par l'algorithme 9 est adaptée pour traiter des problèmes de taille raisonnable, cependant, quand la taille du problème considéré augmente, des difficultés d'implémentation surgissent.

Premièrement, la mémoire nécessaire pour stocker les valeurs de la fonction Q pour tous les couples état-action visités explose rapidement et deuxièmement, il est possible que le temps nécessaire pour explorer tous les couples état-action afin de caractériser correctement l'évolution de la fonction Q du système ne soit trop important.

Cette deuxième difficulté est particulièrement problématique dans les problèmes à espace d'état continu où il est alors impossible d'expérimenter un nombre infini de fois tous les couples état-action. Pour rappel, si l'algorithme d'apprentissage n'expérimente pas plusieurs fois un même état, les valeurs apprises de la fonction Q ne seront pas représentatives du système étudié.

Pour ces raisons, l'utilisation d'une fonction d'approximation est régulièrement mise en œuvre dans les problèmes où une implémentation tabulaire classique n'est pas envisageable. Comme il a été montré dans les sections précédentes, une fonction d'approximation telle qu'un réseau de neurones est un outil puissant pour approximer des fonctions non linéaires complexes et stocker un grand nombre d'informations dans les coefficients synaptiques.

4.5.1 Exemple pratique des limites d'une implémentation tabulaire

Dans les travaux de [130], les auteurs ont pour ambition d'appliquer un processus d'apprentissage par renforcement à un robot qui recherche une source de lumière : dans un environnement 2-D possédant une source de lumière, un robot-agent tente de développer une stratégie efficace pour trouver l'emplacement de cette source de lumière.

L'agent reçoit ainsi des récompenses positives croissantes lorsqu'il se rapproche de la source de lumière et des récompenses négatives croissantes lorsqu'il s'en éloigne. Deux discrétisations de l'espace d'état sont effectuées pour simuler un problème de faible dimension et un problème de plus grande dimension.

Dans une première approche, un algorithme de Q-learning est implémenté, où les valeurs de la fonction Q sont stockées dans une table. Puis, le même algorithme est développé avec un stockage des résultats dans un réseau de neurones. Les différentes expériences ont montré la supériorité de l'implémentation tabulaire pour le problème de faible dimension alors que l'implémentation neuronale est plus efficace quand la dimension du problème est plus importante.

Bien que le problème explicité dans [130] soit assez simpliste, ces travaux constituent une bonne illustration des limites de l'implémentation tabulaire lorsque la dimension de l'espace d'état augmente.

Ainsi, dans la suite des travaux la vision tabulaire de l'apprentissage par renforcement est abandonnée au profit d'une approche utilisant une fonction d'approximmation, plus connue sous le nom d'*approche connexionniste*.

4.5.2 Approche connexionniste

Dans le cadre d'un apprentissage par renforcement mettant en œuvre un réseau de neurones, l'objectif est d'utiliser le réseau pour effectuer une approximation de quelques unes des caractéristiques essentielles déduites par AR afin d'appliquer une politique décisionnelle au problème considéré.

Généralement, une approximation de fonction est utilisée pour estimer les valeurs de $Q^{\pi}(s,a)$, de cette manière n'importe quel algorithme d'apprentissage par renforcement peut être mis en œuvre sans nécessiter de grosses modifications structurelles.

Dans ces travaux, nous nous intéressons à l'application d'une fonction d'approximation à un algorithme de TD-learning classique de type SARSA. L'estimateur a alors pour but principal d'apprendre la récompense[10] reçue après avoir effectué l'action a dans l'état s en suivant la politique π. Ainsi, le couple $\{s,a\}$ représente les entrées de l'estimateur, tandis que $Q^{\pi}(s,a)$ est la sortie souhaitée.

La structure de l'estimateur peut être déterminée assez librement. En effet, certains travaux utilisent les résultats de l'estimateur pour déterminer la meilleure action à effectuer [129], [130], tandis que d'autres entraînent un estimateur par action possible afin d'étudier chaque action indépendamment des autres [194], [195].

L'une des difficultés majeures de l'approche connexionniste est de savoir comment effectuer l'entraînement du réseau de neurones pour apprendre la politique souhaitée. Dans l'approche tabulaire, l'apprentissage s'effectue en ré-écrivant la valeur de $Q^{\pi}(s,a)$ dans la table de stockage, cependant la manœuvre est plus délicate dans le cas d'un approximateur neuronal puisqu'il est nécessaire de conserver les propriétés de généralisation et la précision sur l'estimation fournie.

La méthode la plus communément utilisée consiste à n'entraîner l'estimateur que pour les couples état-action visités par l'agent, afin d'éviter l'apprentissage de cas non utiles à la résolution du problème. En outre, la discrétisation de l'espace d'état est également primordiale pour garantir de bonnes propriétés d'approximation et de généralisation.

4.5.3 Discrétisation de l'espace état-action

4.5.3.1 Malédiction de la dimension

Dans l'éventualité d'un espace état-action continu, il peut s'avérer nécessaire d'effectuer une discrétisation pour pouvoir représenter l'espace d'état de manière synthétique. Or, le coût de discrétisation de l'espace d'état croît exponentiellement avec la dimension de cet espace, ce qui a pour effet majeur de mener à une explosion de la complexité de résolution du PDM. Ce phénomène est connu sous le nom de *malédiction de la dimension* (ou *curse of dimensionality* en anglais).

En d'autres termes, un compromis doit être effectué entre précision et rapidité de calcul : plus la description des états du système est précise, plus le temps de calcul

10. Ici, l'utilisation du terme *récompense* est un abus de langage, puisqu'il s'agit plutôt de déterminer la profitabilité d'une action effectuée dans un état. Néanmoins, cette profitabilité est directement déduite des récompenses reçues successivement par l'agent.

nécessaire à la résolution du problème est important.

Le raisonnement inverse amène alors à la constatation suivante : pour un problème de dimension donné, le nombre de variables utilisées pour décrire un état doit être suffisamment faible pour ne pas rendre la résolution trop coûteuse en temps de calcul.

Dans la littérature, ce compromis est trouvé en employant des méthodes de discrétisation plus appropriées qu'une simple maillage uniforme de l'espace d'état. [196] et [197] proposent par exemple de réduire la densité du maillage de l'espace d'état dans les zones où une plus faible précision est requise. [190], [198] et [199] proposent quant à eux d'utiliser un codage en grille, dont le principe est d'appliquer un certain nombre de maillages imbriqués pour subdiviser l'espace d'état de l'environnement.

4.5.3.2 Discrétisation de l'espace d'état

La discrétisation de l'espace d'action n'est pas nécessaire puisque le nombre d'actions que peut effectuer un agent est relativement faible : dans le cas de l'étude des temps d'arrêt en station pour une ligne de métro, la précision requise est de l'ordre de la seconde. L'espace d'action A est donc naturellement discret et le cas d'un espace d'action continu n'est pas considéré.

En revanche la discrétisation de l'espace d'état est une nécessité puisque celui-ci est continu : lors de l'exploitation, les trains peuvent occuper une infinité de positions possibles.

La discrétisation de l'espace d'état concerne principalement deux notions : le pas de discrétisation et l'encodage des variables.

Le choix du pas de discrétisation permet de fixer la granularité de l'étude. Dans un problème à espace d'état continu, un pas de discrétisation faible entraîne une meilleure précision sur les données au détriment de la taille de l'espace d'état.

L'encodage des variables est également une notion importante, puisqu'il permet à l'estimateur de faire la distinction entre les différents couples état-action qui lui sont présentés, tout en lui permettant de développer ses capacités de généralisation.

De fait, l'encodage doit adopter une certaine logique pour que les états proches dans la réalité bénéficient d'un encodage mettant en évidence leurs points communs.

Cependant, comme le rappelle [185], la méthode de représentation des entrées est largement dépendante du problème étudié, ce qui impose d'effectuer divers essais et erreurs pour déterminer l'encodage le plus approprié.

Par conséquent, une discrétisation dérivée du codage en grille est utilisée, en exploitant les délimitations naturelles imposées par la présence de stations à des positions fixes.

Un positionnement relatif des trains a été privilégié par rapport à un positionnement absolu afin de faire chuter considérablement le nombre de variables nécessaires pour décrire l'état des trains sur la ligne.

Avec ce procédé de discrétisation, seules trois variables sont nécessaires pour représenter efficacement le positionnement d'un train sur la ligne.

4.5.4 Algorithme connexionniste d'apprentissage par renforcement

Les traces d'éligibilité sont utilisées pour simuler une mémoire des expériences afin d'identifier quels couples état-action visités précédemment sont responsables de la récompense reçue au pas de temps courant.

Dans une approche connexionniste, les traces ne sont plus définies par couples état-action, mais par poids synaptiques : la trace d'éligibilité d'un coefficient synaptique donne alors une indication sur l'impact de chaque coefficient dans la mise à jour de l'algorithme de rétropropagation.

Plusieurs difficultés d'implémentation surviennent.

D'une part, l'implémentation connexionniste est plus contraignante que l'approche tabulaire puisqu'il est nécessaire de modifier l'algorithme de rétropropagation de l'erreur pour intégrer les traces d'éligibilité : la règle de mise à jour de la trace d'éligibilité à l'itération k est de la forme de (4.43), où $\Delta\omega_Q$ est la matrice des coefficients synaptiques de la fonction d'approximation définissant Q. Il est alors nécessaire de stocker les matrices de coefficients synaptiques aux différentes itérations pour appliquer cette règle de mise à jour, ce qui nécessite d'allouer de la mémoire supplémentaire.

$$e_k = \gamma\lambda e_{k-1} + \Delta\omega_Q \tag{4.43}$$

D'autre part, cette approche n'est pas compatible avec un apprentissage supervisé puisque l'algorithme d'apprentissage supervisé n'utilise pas la récompense courante pour calculer les gradients de l'erreur, mais les récompenses présentes dans la base de donnée. La récompense courante n'est donc propagée aux couples état-action visités précédemment que s'ils se trouvent dans la base d'apprentissage courante [185].

Dans la littérature, il existe quelques exemples de travaux ayant utilisés le principe des traces d'éligibilité dans une approche connexionniste incrémentale comme [191] ou [194]. Pour rappel, l'apprentissage online consiste à présenter chaque cas d'apprentissage un par un, ce qui dans le cas d'une base de données déjà constituée demande plus de temps d'apprentissage qu'une approche offline.

Pour ces raisons, nous avons choisi de poursuivre l'étude dans le cas d'un apprentissage offline.

4.5.4.1 Neural fitted Q-iteration

La méthode *Neural Fitted Q-iteration* (NFQ) a été introduite par [200] pour offrir une alternative à la mise à jour en ligne des coefficients synaptiques d'un réseau neuronal. L'apprentissage est ainsi réalisé en offline en collectant l'ensemble des transitions $\{s, a, s', r\}$ issues de simulations d'interactions entre l'agent et l'environnement.

Cela revient à effectuer un apprentissage par renforcement offline puisque toutes les interactions sont connues au préalable : la référence d'apprentissage de la fonction Q pour chaque cas est calculée de manière déterministe pour être ensuite apprise par le réseau neuronal via un algorithme classique d'apprentissage supervisé.

La méthode NFQ présente l'avantage de mener à une phase d'apprentissage plus courte de par l'intégration d'un apprentissage supervisé. Une architecture d'apprentis-

sage similaire à celle décrite par la figure 4.8 peut alors être utilisée pour mettre en œuvre cette méthode.

Cette méthode constitue une première solution pour effectuer un apprentissage par renforcement, cependant, le désavantage de cette méthode est de ne pas garder une mémoire des états visités. Ainsi, au lieu d'appliquer le principe des traces d'éligibilité, il a été choisi d'explorer une autre solution : la méthode *Dyna*.

4.5.4.2 Architecture Dyna

L'architecture Dyna a été proposée par Sutton dans [201] et mise en œuvre notamment dans une implémentation tabulaire dans [202] et [203] afin d'effectuer un compromis entre les méthodes model-based et model-free. Cette architecture est également appelée *Dyna-Q learning*.

Sutton la définit comme une "*architecture intégrée permettant un apprentissage, une planification et une réaction*" [204].

Comme le souligne [205], l'architecture Dyna utilise un modèle partiel et déterministe du système qui est enrichi par de nouvelles expériences à chaque nouvelle itération de l'algorithme.

Dans [201], Sutton suggère d'apprendre un modèle de l'environnement pour déterminer les valeurs de la fonction de récompense et les états consécutifs à la prise d'action dans un état donné.

Grâce à cette approche, l'agent est en mesure d'une part de connaître la conséquence des actions effectuées, mais également d'effectuer une estimation sur les signaux de renforcement pour les actions non explorées.

D'un point de vue pratique, l'architecture Dyna et les traces d'éligibilité ont le même impact sur l'apprentissage en permettant de mettre à jour plusieurs couples état-action à chaque itération.

Les traces d'éligibilité permettent de garder une mémoire temporaire de la trajectoire visitée alors que l'architecture Dyna permet de développer un modèle de l'environnement, ce qui en définitive crée une mémoire à long terme de toutes les expériences et de leur impact respectif sur l'objectif de l'agent.

Dans [206], les auteurs effectuent une comparaison entre une méthode SARSA(λ) et une méthode Dyna-SARSA dans une implémentation tabulaire d'un problème de labyrinthe. Ils montrent que l'architecture Dyna est plus efficace à trouver la politique optimale du problème dès lors que la taille de l'espace d'état augmente.

Par contre l'architecture Dyna présente de moins bonnes performances qu'une méthode intégrant des traces d'éligibilité dans le cas où les états du système sont partiellement observables.

Un algorithme simplifié de la méthode Dyna-Q est proposé par l'algorithme 10, où χ est le modèle de l'environnement, Φ est la fonction dont dérive la politique π et Ω est le modèle de récompense[11].

Il est à noter que certaines versions de Dyna-Q utilisent un même approximateur pour stocker les couples (s', r), cependant l'utilisation d'une fonction d'approximation

11. les variables P, R et Q introduites précédemment sont respectivement remplacées par χ, Ω et Φ afin de marquer l'originalité de la démarche Dyna-Q connexionniste par rapport à la définition classique d'un PDM.

par variable permet une plus grande souplesse et facilite la phase d'apprentissage dans une approche connexionniste.

La première partie de l'algorithme 10 correspond à une routine d'apprentissage de fonction Q d'une méthode NFQ classique avec une phase d'apprentissage des fonctions χ, Ω et Φ, tandis que la seconde partie permet d'intégrer la notion de planification à la fonction Q en exploitant χ et Ω pour mettre à jour la fonction Φ pour des couples état-action déjà visités.

Le modèle de l'environnement est alors substitué à l'environnement réel pour accélérer le processus d'apprentissage [207].

Notons que si m a une valeur nulle, l'algorithme 10 se résume à un apprentissage par renforcement de type Q-learning. L'étape de planification est donc une étape optionnelle pouvant être activée ou désactivée au gré du processus d'apprentissage.

Algorithme 10 Algorithme type Dyna-Q connexionniste

Initialiser la politique $\pi(\Phi)$ suivie par l'agent
Initialiser les fonctions χ et Ω
Pour tout $a \in A$, $s \in S$ **Faire**
    Initialiser la fonction $Q(s, a)$
Fin du Pour
Pour tout épisodes **Faire**
    Initialiser l'état de départ s.
    Tant que L'état final n'est pas atteint **Faire**
        $a \leftarrow \pi(\Phi)$
        Recevoir la récompense r et observer l'état suivant s'
        $a' \leftarrow \pi(\Phi)$
        Mettre à jour la fonction Φ avec la règle (4.37)
        $\chi(s, a) \leftarrow s'$
        $\Omega(s, a) \leftarrow r$
        $s \leftarrow s'$
        Pour m itérations **Faire**
            $s \leftarrow$ Choisir aléatoirement un état déjà visité s
            $a \leftarrow$ Choisir aléatoirement une action a déjà effectuée en s
            $s' \leftarrow \chi(s, a)$
            $r \leftarrow \Omega(s, a)$
            Mettre à jour la fonction Φ avec la règle (4.37)
        Fin du Pour
    Fin du Tant que
Fin du Pour

Dans une approche connexionniste et en supposant un apprentissage réalisé correctement, il peut être supposé que les fonctions χ, Ω et Φ développent des propriétés de généralisation qui permettent d'aller plus loin dans le processus de planification, en effectuant de l'anticipation, notamment en traitant des couples état-action non visités.

Cette propriété n'est exploitée que très rarement dans la littérature comme dans [208], dans le but de développer les capacités d'anticipation de robots.

Trois raisons principales pourraient expliquer cet état de fait : premièrement, il est nécessaire d'utiliser une approche connexionniste pour réaliser l'apprentissage par ren-

forcement du modèle de l'environnement, deuxièmement, l'apprentissage des fonctions χ, Ω et Φ doit permettre d'atteindre un taux d'erreur d'estimation suffisamment faible pour avoir confiance dans la prédiction réalisée sur de nouveaux événements et troisièmement, le temps de calcul nécessaire pour atteindre ce niveau d'erreur peut s'avérer prohibitif dans le cas de problèmes complexes.

En effet, il peut apparaître difficile d'atteindre un taux d'erreur suffisamment faible dans des problèmes complexes de grande dimension, puisque cela supposerait d'avoir visité suffisamment de couples état-action uniformément répartis dans l'espace un grand nombre de fois.

4.5.4.3 Pourquoi utiliser une architecture Dyna neuronale ?

La réalisation des objectifs initiaux de l'optimisation énergétique en temps réel, nous a amené à considérer les trois caractéristiques suivantes.

Capacité d'approximation : l'utilisation d'un réseau neuronal permet de développer des capacités d'approximation souhaitable dans le cas de problèmes présentant un espace état-action dont l'exploration exhaustive est impossible.

De plus le temps de réponse d'un réseau neuronal est négligeable ce qui en fait un bon candidat pour des applications temps-réel.

Rapidité et convergence de l'apprentissage : l'apprentissage supervisé a démontré des capacités particulièrement utiles pour traiter des problèmes complexes en assurant la convergence de l'apprentissage en un temps de calcul raisonnable.

Modèle de l'environnement et de la fonction récompense : la constitution d'un modèle de l'environnement et d'un modèle de la fonction récompense permet de développer des capacités de planification utiles si l'on souhaite limiter la taille de la base de donnée après avoir effectué l'apprentissage d'une politique optimale.

Une fois ces modèles appris, il peut être envisagé de ne mettre à jour la base de donnée d'apprentissage qu'avec de nouveaux cas d'étude simulés ou mesurés afin d'éviter tout sur-apprentissage.

Une méthode Dyna-Q connexionniste intégre ces trois caractéristiques dans son fonctionnement sous l'hypothèse de réaliser un apprentissage supervisé des fonctions d'approximation, ce qui en fait une technique adéquate pour répondre aux enjeux de l'optimisation temps réel.

L'algorithme d'optimisation hybride développé au chapitre précédent est ainsi mis à profit afin d'ajouter la notion d'apprentissage supervisé à la méthode Dyna-Q pour créer une base de données permettant d'orienter l'apprentissage plus rapidement vers une politique conforme aux objectifs visés.

Cette méthode nommée *Dyna-NFQ* (DNFQ) s'inspire du processus décrit par l'algorithme 10 et y rajoute une base d'apprentissage regroupant un ensemble d'épisodes dont les caractéristiques sont connues.

4.5.5 Méthode Dyna-NFQ

4.5.5.1 Batch training

Contrairement à l'apprentissage réalisé dans le cadre de l'estimateur neuronal des flux de puissances, l'apprentissage par renforcement nécessite une mise en œuvre un peu différente bien que dans les deux cas l'apprentissage soit supervisé, notamment parce qu'ici il est nécessaire d'aller plus loin que le simple apprentissage d'une base de données, il est également nécessaire de développer une politique efficace utilisable pour des cas non présents dans la base de données.

Un apprentissage online peut être envisagé, mais présente le désavantage majeur que lors des premières itérations, la politique construite ne soit pas représentative des données futures, ce qui rend le processus d'apprentissage progressif et donc plus long.
A l'inverse, en connaissant par avance l'ensemble des cas de la base d'apprentissage, il est possible dès les premières itérations de l'apprentissage de construire une politique tenant compte de toutes les caractéristiques connues du système étudié.

En reprenant l'analogie avec un jeu de plateau, un apprentissage offline est assimilable à un apprentissage où l'on connaît d'avance toutes les règles du jeu, tandis que dans un apprentissage online, l'agent apprenant ne reçoit les informations concernant les règles du jeu que progressivement au cours des parties jouées, ce qui bien évidemment rend les politiques apprises lors des premières parties inefficaces.

Malgré tout, la progressivité de l'apprentissage online est une caractéristique intéressante pour effectuer un apprentissage sur une base de données de grande taille. Dans la littérature l'adaptation de cette caractéristique à un apprentissage est appelée *batch training*.
Les données d'apprentissage sont donc partitionnées en sous-groupes (batch) de manière à faciliter l'apprentissage. Chaque sous-groupe de données est présenté itérativement au réseau neuronal jusqu'à ce que l'erreur d'estimation atteigne un niveau acceptable.
Enfin, le batch training permet au réseau neuronal de passer régulièrement en revue des cas appris précédemment pour ne pas que l'agent apprenant oublie ce qu'il a déjà appris.

4.5.5.2 Hint to the goal

Dans le cadre d'un apprentissage par renforcement, l'apprentissage des cas défavorables est presque aussi important que celui des cas favorables, puisqu'il est souhaitable que durant la phase d'exploitation l'agent puisse choisir des actions permettant de maximiser les récompenses reçues, mais également que durant la phase d'exploration, il puisse faire l'expérience de "mauvaises" situations soit pour les éviter par la suite, soit pour les utiliser afin de se retrouver dans un état plus favorable dans les transitions suivantes.

En outre, dans le cas de l'apprentissage supervisé classique d'un réseau neuronal, Riedmiller évoque l'utilisation de cas artificiels d'apprentissage pour forcer le réseau neuronal à reconnaître les états terminaux (ou états favorables) en leur assignant une valeur de coût en conséquence [200], [209].

L'auteur donne à cette méthode le nom de *hint to the goal*, puisqu'elle permet d'indiquer une zone de l'espace comme la région à atteindre pour réaliser l'objectif de l'apprentissage.

Le principe inverse peut donc être en théorie applicable pour forcer le réseau neuronal à reconnaître les états défavorables ou ne menant pas aux récompenses maximales.

Cela consiste à générer des cas d'apprentissage sur la base d'interactions simulées d'un agent avec son environnement et à ne pas écarter les solutions jugées comme mauvaises afin de s'assurer que la base d'apprentissage contienne à la fois des indications sur les *bonnes* et *mauvaises* situations.

En pratique, ces mauvaises solutions sont obtenues lors des premières itérations de l'optimisation hybride OEP-AG (cf. figure 3.4.8).

4.5.5.3 Observations empiriques

En outre, la réalisation de multiples simulations d'apprentissage a permis de mettre en lumière certaines contraintes empiriques :

— L'augmentation de la taille du cache doit être progressive au cours des itérations. Un cache est défini comme la liste de données qui est réellement présentée à l'agent apprenant.

  Ainsi, en fin de cycle d'apprentissage, la taille du cache est égale à celle du batch. L'utilisation d'un cache dont la taille est incrémentée au fil des itérations permet d'assurer la progressivité de l'apprentissage afin d'éviter les modifications trop importantes de la politique apprise.

— Une règle de remplacement des données d'apprentissage doit être appliquée lorsque la taille maximale de la base de donnée est atteinte.

  Malgré les progrès technologiques de ces dernières années sur les coûts du stockage de données et de la puissance de calcul, il est toujours nécessaire de limiter la taille de la base d'apprentissage pour s'assurer que l'apprentissage soit réalisé en un temps raisonnable.

  La forme de la règle de remplacement (4.44) s'inspire de celle développée par [210].

  Cette règle de remplacement insère un nouveau cas dans la base d'apprentissage en remplaçant le cas le plus similaire à ce nouveau cas. ξ_1 et ξ_2 sont respectivement les coefficients de pondération permettant de régler l'importance de la distance entre les états s et les actions a.

$$\vartheta = \sqrt{\xi_1(a_{new} - a_{old})^2 + \xi_2(s_{new} - s_{old})^2} \tag{4.44}$$

— Il a été constaté qu'en ordonnant les données contenues dans le cache avant de les faire apprendre au RNA l'erreur d'apprentissage décroissait plus rapidement qu'en effectuant un batch learning aléatoire.

— Un réapprentissage ponctuel des cas menant aux états les plus favorables permet également de guider la prise de décisions au fil des itérations vers une politique maximisant les récompenses.

4.5.5.4 Implémentation de la méthode Dyna-NFQ

Les macro-étapes de la mise en œuvre de la méthode Dyna-NFQ sont décrites par la figure 4.23. La première étape consiste à créer une base de données contenant les caractéristiques $\{s, a, r, s'\}$ d'un grand nombre d'épisodes.

Ensuite, les couples $\{s, a\}$ sont utilisés pour entraîner les fonctions d'approximation χ, Ω et Φ. Les références s'_{ref} et r_{ref} sont directement issues de la base de données, tandis que Q_{ref} est approché par un critère de récompense actualisé similaire à (4.27).

L'apprentissage supervisé des fonctions d'approximation χ, Ω et Φ est réalisé en appliquant les préconisations de partitionnement de la base d'apprentissage afin de réduire au maximum le temps de convergence des fonctions vers une erreur d'estimation faible et la définition d'une politique optimale.

En outre, la méthode DNFQ étant basée essentiellement sur un apprentissage supervisé d'une base de données, la majorité des points explicités dans la partie concernant la conception d'un estimateur neuronal peuvent être repris comme la normalisation des données d'apprentissage, le suivi de l'erreur d'apprentissage ou encore l'étude des performances des fonctions d'approximation.

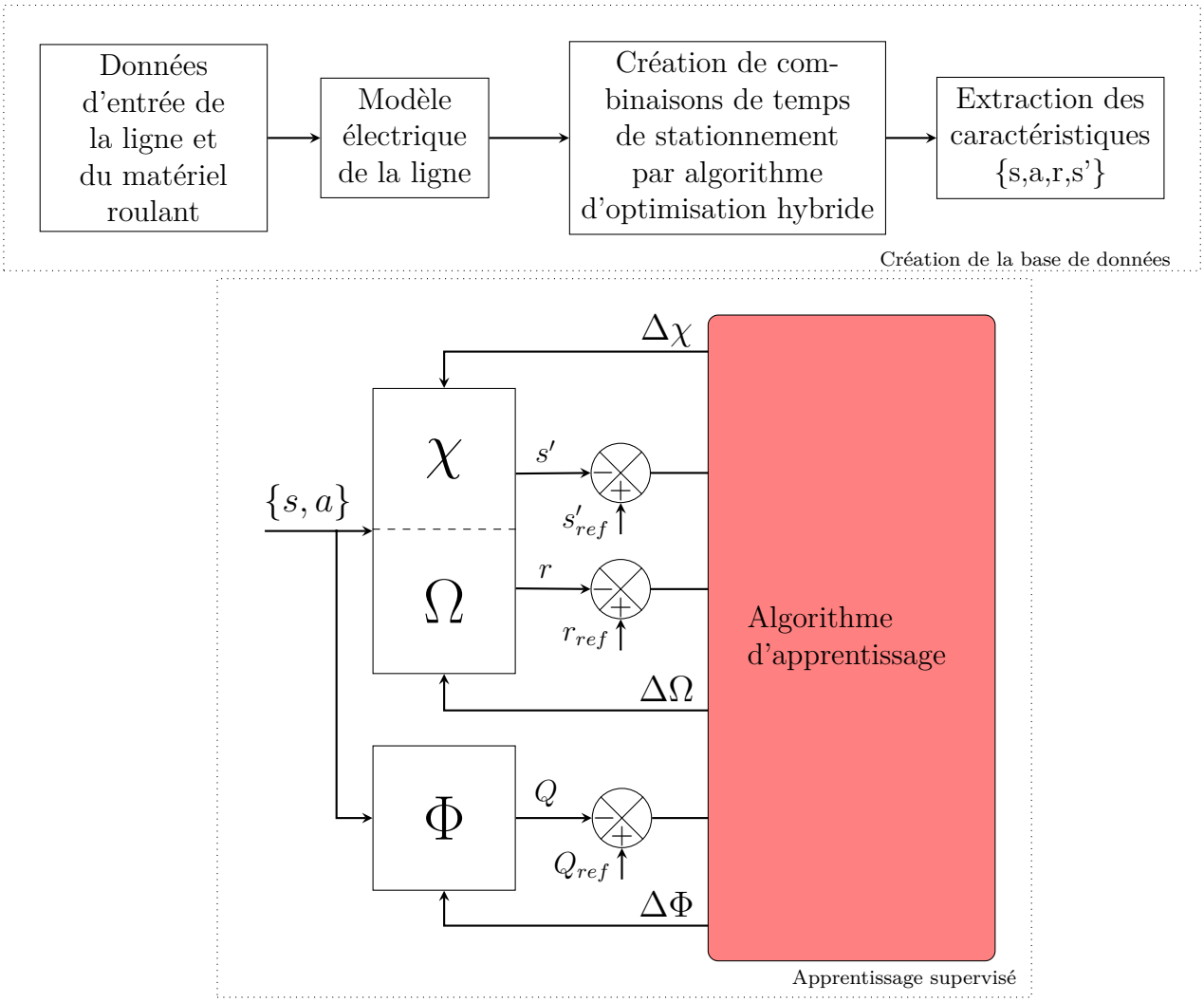


FIGURE 4.23 – Structure de la méthode Dyna-NFQ.

4.5.6 Robustesse de la méthode face aux perturbations

4.5.6.1 Étude des aléas de trafic

La base d'apprentissage générée par l'algorithme d'optimisation hybride joue un rôle prépondérant dans le fonctionnement de la méthode DNFQ. Elle doit être suffisamment représentative du comportement moyen de la ligne de métro pour s'assurer que la fonction valeur et le modèle de l'environnement appris restent vrais dans le cas de perturbations.

Par extension, il peut être intuitivement spécifié que plus le nombre de cas de simulations est important plus il y a de chance de rencontrer des cas d'exploitation qui pourraient survenir en exploitation réelle, sous l'hypothèse de laisser aux algorithmes d'optimisation une marge de manœuvre suffisamment importante pour simuler des cas d'exploitation intégrant des aléas.

Les différents enregistrements réalisés lors d'essais-site sur la ligne de Turin ont permis de mener une étude statistique de la nature des aléas d'exploitation les plus probables et de leur fréquence moyenne d'occurrence. Ces résultats sont synthétisés par les figures 4.24, 4.25 et 4.26.

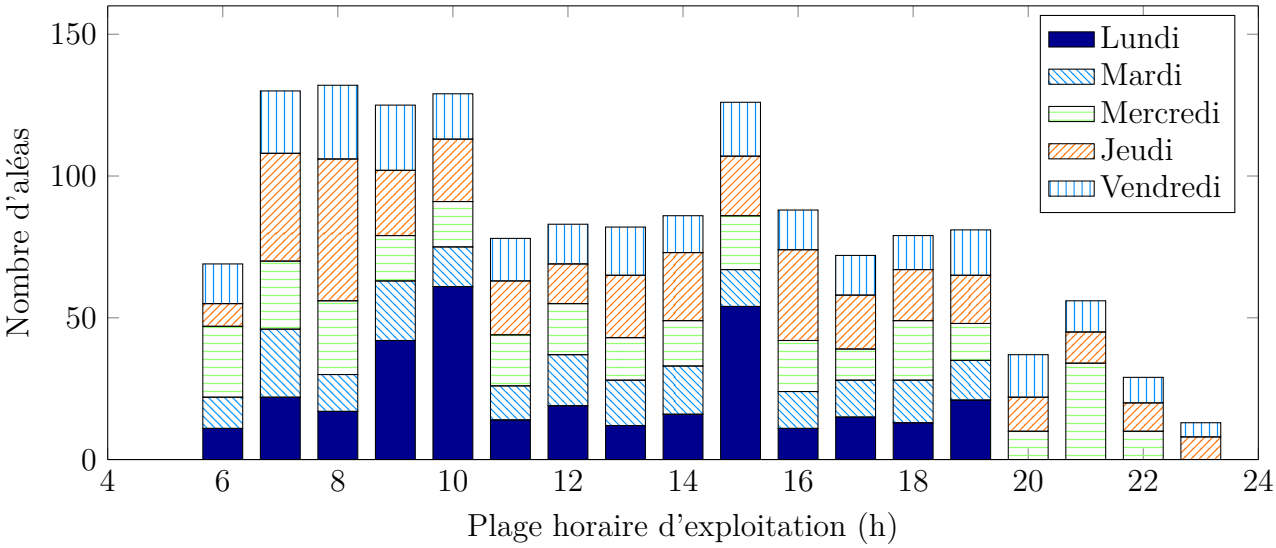


FIGURE 4.24 – Distribution des aléas hebdomadaires en fonction de la plage horaire d'exploitation

D'après la figure 4.26, environ 75% des aléas hebdomadaires ont une durée inférieure ou égale à 2s.

Ainsi, le paramétrage de l'algorithme d'optimisation hybride doit être défini de manière à explorer des solutions représentatives de la distribution des aléas afin de pouvoir recréer une dispersion des temps de stationnement analogue. De cette manière, la méthode DNFQ est en mesure d'intégrer ces conditions de fonctionnement dans les modèles appris et de développer une politique robuste permettant de re-synchroniser les trains dans n'importe quelle configuration de carrousel.

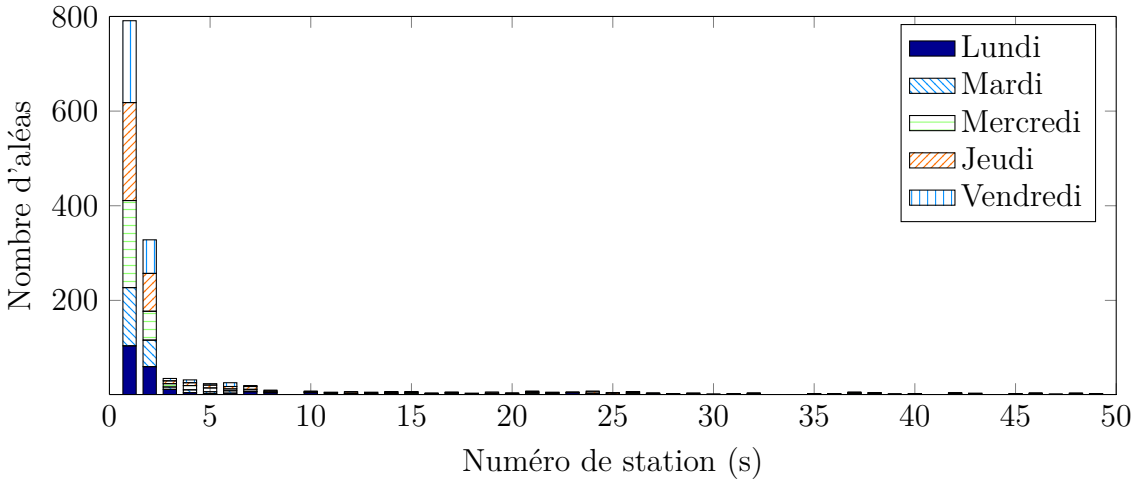


FIGURE 4.25 – Distribution de la durée des aléas d'exploitation hebdomadaires

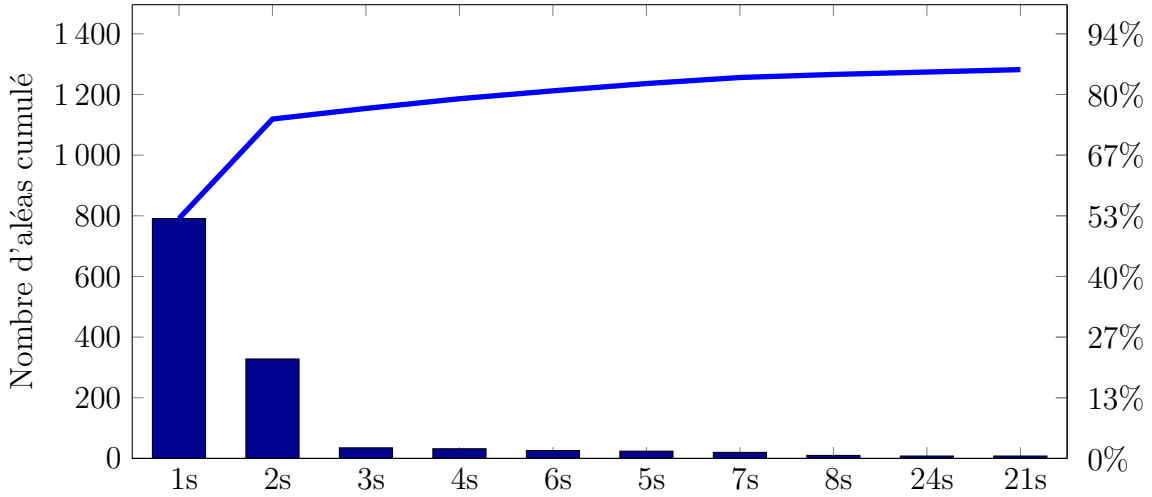


FIGURE 4.26 – Distribution de pareto de la durée des aléas hebdomadaires

4.5.6.2 Étude de robustesse

Dans [211], les réseaux de neurones sont définis comme tolérants aux fautes ou robustes de par leurs propriétés intrinsèques. Dans notre cas d'étude, la propriété la plus intéressante est *la nature distribuée de l'information* : du fait du grand nombre de neurones dans un réseau, chaque neurone contribue dans une faible proportion à la réponse finale. De cette propriété découle la capacité de généralisation des RNA [212].

Dans [211], il est fait mention de deux types de fautes : une dégradation interne des informations contenues dans le RNA (une perte d'information sur les poids synaptiques ou la suppression d'un neurone) et le bruit des données d'entrées. C'est ce dernier type de fautes qui est le plus susceptible de se produire puisque les données d'entrées présentées au RNA sont dépendantes des aléas de trafic.

En outre, comme le montre [213], la robustesse d'un réseau de neurones est améliorée en lui présentant des cas d'apprentissage entachés de bruit.

La robustesse de la politique est alors obtenue en considérant suffisamment de cas de fonctionnement dégradé de la ligne de métro pour que ces points de fonctionnement

soient considérés comme des cas de fonctionnement nominaux.

Pour éprouver la robustesse de la méthode deux cas d'études sont considérés : un cas d'exploitation en heure creuse et un cas d'exploitation en heure de pointe.

Les fréquences ϑ_f et amplitudes ϑ_a des aléas dans chacun des deux cas sont définies selon les conditions d'exploitation moyennes enregistrées lors des essais sur la ligne de Turin aux périodes correspondantes.

La figure 4.27 présente la distribution de l'amplitude ϑ_a des aléas dans chacun des deux cas. Les fréquences d'occurrence des aléas sont respectivement $\vartheta_{f_1} = 260s$ $\vartheta_{f_2} = 145s$.

Ces deux cas de figures sont étudiés pour un même carrousel avec des conditions initiales identiques afin d'observer les performances de la méthode DNFQ lorsque la fréquentation de la ligne varie.

En heure de pointe, du fait d'une fréquentation de la ligne plus importante, les aléas sont plus récurrents qu'en heure creuse, ce qui correspond à un cas défavorable pour la méthode DNFQ.

En outre, l'écart-type de l'amplitude des aléas en heure de pointe est volontairement augmenté pour simuler des perturbations telles que l'algorithme doit suivre la consigne de temps de stationnement minimal pour résorber le retard des trains concernés par les aléas. Ainsi, le retour à un état optimisé doit être coordonné en modifiant les temps de stationnement de plusieurs trains arrivés en station consécutivement.

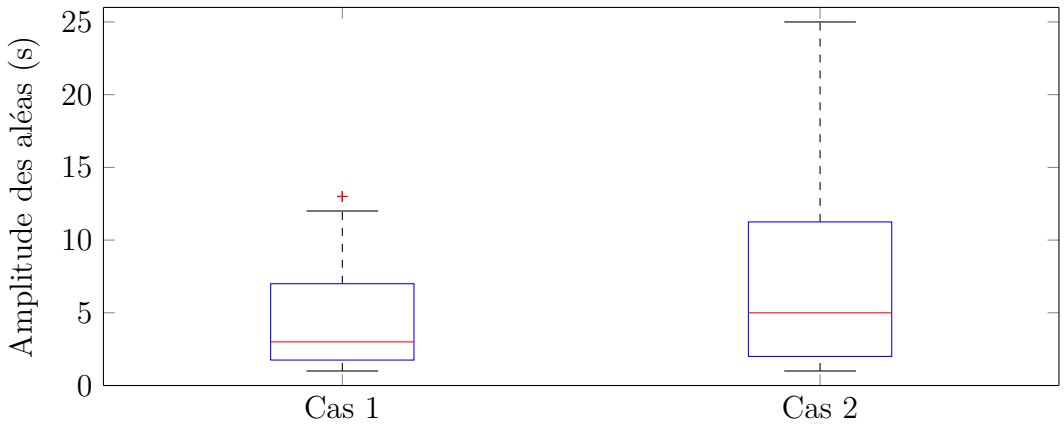


FIGURE 4.27 – Distribution des amplitudes des aléas dans les deux cas d'études.

Cette comparaison permet ainsi d'évaluer la capacité de la méthode à pouvoir redéfinir la politique décisionnelle grâce à la phase de planification : les modèles χ et Ω du système sont utilisés pour créer de nouveaux cas d'apprentissage tenant compte de la nouvelle distribution des aléas afin de modifier la politique Φ.

4.5.6.3 Performances de la méthode DNFQ

La base de données d'apprentissage est constituée en appliquant la méthode d'optimisation hybride OEP-AG. La distribution des solutions obtenues est illustrée par la figure 4.28.

Une très forte densité de solutions générant un gain énergétique supérieur à 10% a été produite en laissant l'algorithme d'optimisation évoluer sur un grand nombre

d'itérations, cela permet ainsi d'orienter l'apprentissage d'une politique menant à des décisions favorables pour la réutilisation de l'énergie issue du freinage récupératif.

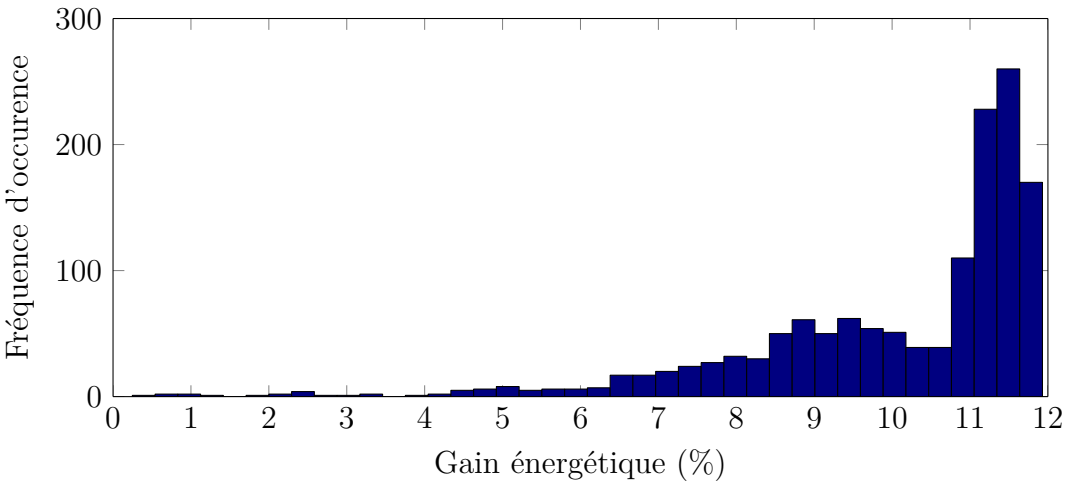


FIGURE 4.28 – Distribution des solutions constituant la base d'apprentissage.

La figure 4.29 présente l'évolution de la consommation énergétique du carrousel sur un tour de boucle dans chacun des deux cas d'études. A chaque itération, la politique courante est appliquée pour un carrousel effectuant un tour de boucle complet. De plus, un même profil d'occurrence des aléas est employé à chaque itération.

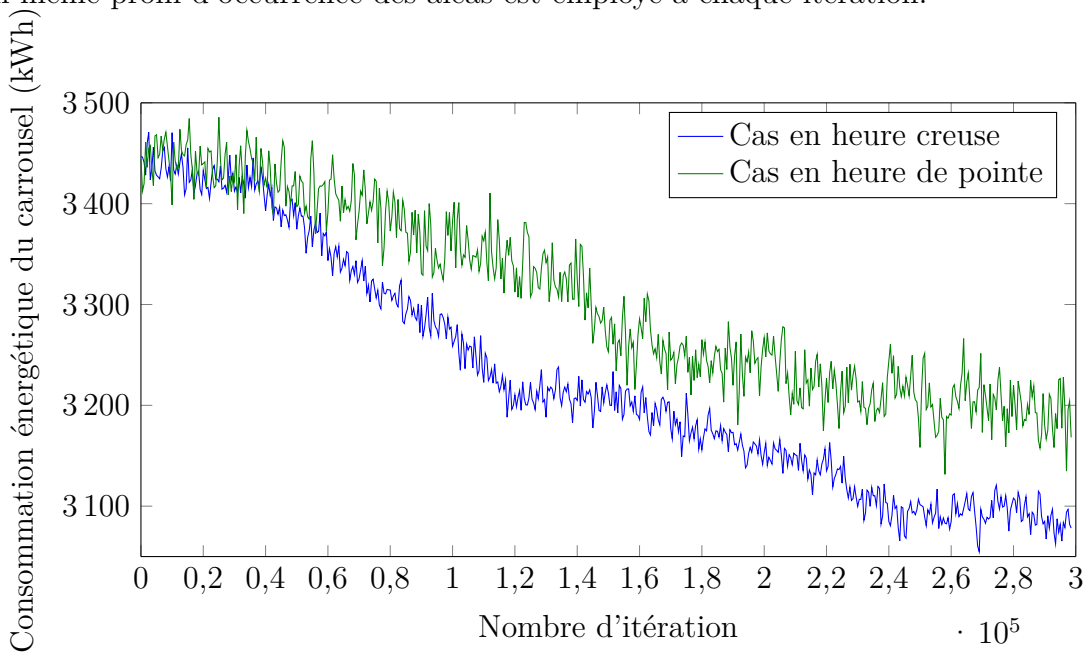


FIGURE 4.29 – Évolution de la consommation énergétique d'un carrousel induite par la politique de stationnement suivie dans deux cas d'études.

Les performances de la méthode DNFQ dans chacun des deux cas d'études de perturbations sont synthétisées dans le tableau 4.4.

Le cas de référence correspond à une exploitation où les trains circulent sans aléas et en suivant la table horaire nominale. Il ne s'agit donc pas d'un cas d'exploitation optimisée, mais de la consommation de base d'un carrousel suivant la consigne nominale initiale fournie par l'exploitant.

Concrètement, il s'agit de la consommation énergétique d'un carrousel sur un tour de ligne en suivant une table horaire nominale et en considérant un cas d'exploitation idéal.

Les autres cas d'études prennent ce cas comme référence dans le calcul des gains énergétiques.

	cas de référence	**cas 1**	**cas 2**
Gain énergétique final	0%	10,3%	7.5%
Temps moyen de prise de décision	✗	$0,468ms$	$0,512ms$
Temps total d'apprentissage	✗	94h15	95h02

TABLEAU 4.4 – Comparaison des performances de la méthode DNFQ dans 2 cas d'exploitation avec perturbations.

Concernant les cas d'étude liés à l'utilisation de la méthode DNFQ (cas 1 et cas 2), l'apprentissage a été limité à un nombre fini d'itérations afin de garder un temps d'apprentissage inférieur à 96h, mais également pour être en mesure de comparer l'évolution de la politique dans les deux cas de figures.

Les temps de calcul correspondent à un PC embarquant un processeur Intel Xeon W3520 cadencé à 2.67GHz, 12 Go de RAM, une carte graphique Quadro FX 3800 avec 1Go de VRAM et le tout sous un environnement Windows 7. Notons que le temps de prise de décision dans le cas du cas de référence n'est pas spécifié, puisque dans ce cas, la décision est connue en amont de la simulation.

Dans le cas où l'exploitation est perturbée par de faibles aléas, le gain énergétique généré par la méthode DNFQ est de 10,3%, tandis que lorsque les perturbations de trafic sont 80% plus fréquentes et de plus grande amplitude, le gain énergétique généré est de 7,5%.

D'après l'évolution des courbes de la figure 4.29, il semble que les performances des politiques décisionnelles issues de la méthode DNFQ pourraient encore être améliorées en permettant une phase d'apprentissage encore plus conséquente.

4.5.6.4 Comparaison par rapport à l'optimisation hors-ligne

Les résultats présentés dans le tableau 4.4 sont à contraster par rapport aux performances d'une table horaire figée dans le cas d'une exploitation soumise à des aléas.

Deux nouveaux cas d'études sont donc définis pour étudier l'impact de l'utilisation de table horaire fixe optimisée pour gérer des aléas en suivant les distributions de perturbations analogues aux cas 1 et 2.

L'inconvénient majeur de cette approche est que selon les conditions de perturbations, une table horaire peut naturellement permettre de résorber les aléas ou favoriser les phases de synchronisation entre accélération et freinage des trains.

Ainsi, en préambule de cette étude, les 50 meilleures tables horaires optimisées issues de la base d'apprentissage (figure 4.28) sont sélectionnées. Puis les gains énergétiques issus de l'application de ces tables horaires dans les deux conditions de perturbations sont analysés.

Ces résultats sont ensuite moyennés afin d'évaluer le comportement énergétique moyen de tables horaires optimisées en hors-ligne dans un environnement dynamique

soumis à perturbations [12].

Les résultats de cette étude sont repris dans le tableau 4.5, les cas 3 et 4 correspondent respectivement aux conditions de perturbations des cas d'étude 1 et 2.

	cas de référence	**cas 3**	**cas 4**
Gain énergétique moyen	0%	3,2%	0,9%
Gain énergétique maximal	0%	6,4%	2,2%
Gain énergétique minimal	0%	-1,3%	-4,1%

TABLEAU 4.5 – Comparaison des performances de l'optimisation hors-ligne dans 2 cas d'exploitation avec perturbations.

Dans le tableau 4.5, l'explicitation des gains moyens et extrêmes permet de montrer que des tables horaires optimisées en hors-ligne (générant des gains énergétiques élevés dans des cas idéaux d'exploitation) peuvent entrainer une augmentation de la consommation énergétique globale de la ligne par rapport à l'utilisation d'une table horaire nominale, lorsque des aléas d'exploitation surviennent.

Ainsi, de cette étude, il est assez compliqué d'estimer l'impact d'une optimisation hors-ligne sur un problème dynamique temps-réel puisque les gains énergétiques perçus dépendent de la table horaire initiale utilisée. En effet, contrairement à une méthode mathématique fournissant un même optimum global à chaque évaluation, la méthode hybride OEP-AG fournit à chaque évaluation un optimum différent.

Cependant, l'analyse des tableaux 4.4 et 4.5 indique d'une part que l'utilisation d'une optimisation hors-ligne pour solutionner un problème temps réel s'avère aléatoire puisque des gains négatifs peuvent être obtenus, et d'autre part, ce type d'approche est moins efficace qu'une approche de type DNFQ, en considérant l'espérance de gain énergétique pour chaque approche.

4.6 Conclusion

Dans le chapitre précédent, une méthode d'optimisation hybride a été définie pour effectuer la modification des temps de stationnement dans un cas idéal d'exploitation. Dans ce chapitre, nous souhaitions donc adapter cette méthode d'optimisation pour effectuer une resynchronisation en temps réel des trains.

Il était ainsi nécessaire de diminuer le temps de calcul de la boucle d'optimisation tout en conservant sa précision.

Un état de l'art des travaux sur la replanification temps réel dans le domaine ferroviaire a mis en évidence qu'il est indispensable de disposer de modèles énergétiques précis de la consommation des trains pour assurer l'optimalité de la prise de décision.

Or, le temps de calcul de la boucle d'optimisation est expliqué à 90% par le calcul des flux de puissances vu par trains à l'aide d'une méthode de résolution itérative pour connaître la consommation énergétique de la ligne.

12. Pour plus de réalisme, lorsque l'avance/retard des trains par rapport à l'horaire initial est trop important, des règles de régulation sont appliquées pour modifier les valeurs nominales de temps de stationnement afin de ne pas violer les contraintes d'exploitation. Ces règles sont définies de manière à garantir un temps de battement minimal.

De fait, la première partie de ce chapitre a été dédiée à la synthèse du processus de résolution itératif par un réseau de neurones artificiels (RNA). Les RNA ont démontré dans un grand nombre d'applications leur capacité à apprendre le fonctionnement d'un système et de fournir une réponse à une sollicitation en un temps réduit.

Après avoir rappelé la théorie mathématique sur les réseaux de neurones et défini les paramètres d'implémentation, les performances de l'estimateur neuronal ont été testées pour des bases de données composées respectivement de 3.10^3 et 4.10^4 cas d'apprentissage.

Le temps de calcul est amélioré d'un facteur 130 dans le premier cas d'étude, contre un facteur 75 dans le deuxième cas de figure. La précision de l'estimateur est quant à elle dépendante du temps alloué à l'apprentissage du RNA, ainsi plus l'apprentissage est long et plus l'estimateur neuronal est capable d'affiner sa précision et d'augmenter ses capacités de généralisation.

Ensuite, il a été décidé d'appliquer le même principe pour effectuer l'apprentissage de la méthode d'optimisation hybride OEP-AG. La méthodologie de réalisation est un peu différente de celle employée pour concevoir un estimateur neuronal, puisque cette fois, nous souhaitons non seulement définir une politique décisionnelle en fonction des cas d'apprentissage issus de l'optimisation hors-ligne, mais également que cette politique puisse s'adapter en cas de perturbations de trafic.

Pour ce faire, nous avons choisi de mettre en œuvre une méthode basée sur l'apprentissage par renforcement, qui consiste à distribuer des récompenses en fonction des décisions prises. Ainsi, la politique décisionnelle optimale est obtenue en effectuant les actions qui maximisent les récompenses reçues sur un horizon temporel.

Une étude de différentes applications utilisant le principe d'apprentissage par renforcement nous a permis de définir la méthode DNFQ, qui est une méthode hybride faisant intervenir l'apprentissage supervisé de plusieurs RNA utilisés conjointement pour générer de nouveaux cas d'apprentissage afin de développer les capacités de planification de la politique décisionnelle.

Les performances de la méthode DNFQ ont ensuite été testées pour deux distributions d'occurrence d'aléas. Ces distributions sont conçues de manière à simuler un cas présentant des aléas de faibles amplitudes et un deuxième cas où les perturbations ont une amplitude et une fréquence beaucoup plus importantes. A l'issue de la phase d'apprentissage, dans le premier cas de figure, la méthode DNFQ génère une politique permettant un gain énergétique de 10,3%, tandis que dans le deuxième cas, le gain constaté est de 7,5%.

En outre, le temps de prise de décision est de l'ordre de 0,5 ms, ce qui rend possible l'implémentation de cette méthode pour une application temps réel.

Ces résultats sont ensuite contrastés par l'étude des performances de tables horaires issues d'une optimisation hors-ligne pour effectuer la gestion énergétique d'une ligne dans des conditions réelles d'exploitation. Il en ressort que ce type d'approche n'est pas déterministe puisque la consommation énergétique globale d'une ligne peut ainsi se retrouver dégradée par rapport à l'utilisation d'une table horaire nominale.

Chapitre 5

Conclusions et perspectives de l'étude

Sommaire

5.1 Conclusions générales

Les différentes méthodes développées dans les chapitres de cette thèse ont permis d'implémenter un outil d'aide à la décision permettant de resynchroniser en temps réel les trains de manière à limiter la dissipation de l'énergie issue du freinage.

Un modèle énergétique d'une ligne de métro a d'abord été défini, puis une méthode de résolution itérative a été décrite pour pouvoir calculer les flux de puissances effectués entre les trains et les sous-stations.

Ensuite, une méthode d'optimisation hybride a été développée pour déterminer des combinaisons de temps d'arrêt en station permettant de minimiser la consommation énergétique.

Puis, la méthode de résolution itérative a été synthétisée par un réseau de neurones afin de gagner en temps de calcul tout en conservant la précision sur les flux de puissances réels.

Enfin, la méthode d'optimisation a été utilisée pour générer une base de données qui a ensuite été apprise par un algorithme DNFQ afin d'en déduire une politique décisionnelle optimale en temps réel. Des simulations réalisées sur la méthode DNFQ pour deux conditions d'exploitation avec aléas ont montré des gains énergétiques potentiels compris entre 7,5% et 10,3%.

Les développements futurs de ces travaux de thèse pourraient consister d'une part à développer une modélisation de lignes de métro encore plus précise afin de se passer de la nécessité de posséder des enregistrements réels pour en déduire une politique optimale pertinente, et d'autre part à développer une méthode d'optimisation mathématique des temps de stationnement pour s'assurer de constituer une base de données présentant des solutions optimales.

5.2 Perspectives

5.2.1 Perspectives d'efficacité de la méthode

5.2.1.1 Limites de la méthode

La méthode DNFQ a été testée sur un cas d'étude relativement simple qui consiste en un tour de boucle d'un carrousel établi. Pour la mettre en œuvre, trois éléments sont indispensables :

— Un estimateur des flux de puissance déduit de la méthode de résolution itérative. Pour chaque carrousel, il est nécessaire d'entraîner un estimateur dédié puisque le nombre d'entrées et de sorties varie en fonction du nombre de trains en ligne.

— Une base de données contenant des solutions potentielles du problème d'optimisation des tables horaires. Cette base est obtenue par un algorithme d'optimisation hybride OEP-AG.

— Un modèle de l'environnement et un modèle de politique optimale issus de l'application de la méthode DNFQ. Ces modèles sont également dépendants du nombre de trains en exploitation. Il est donc nécessaire d'effectuer un apprentissage pour chaque carrousel.

Ainsi, pour appliquer la méthode DNFQ sur une journée d'exploitation, il est indispensable d'avoir complété ces trois étapes pour chaque carrousel exploité lors de la journée, ce qui nécessite un temps d'apprentissage assez conséquent.

En outre, la modélisation mécanique du matériel roulant représente la plus grande source d'incertitude. De fait, pour réduire cette erreur, une instrumentation de rames en exploitation permettrait d'obtenir des données beaucoup plus réalistes concernant le fonctionnement énergétique de la ligne de métro.

5.2.1.2 Gains énergétiques potentiels

Une étude de la répartition des flux énergétique nécessaires à l'avancement des trains se basant sur les travaux de [56], [214], [215], des documents internes à Siemens ainsi que les essais-sie réalisés sur la ligne de Turin a permis de déterminer la figure 5.1.

Cette figure recense et quantifie l'ensemble des pertes énergétiques dans une ligne de métro, ainsi que la quantité d'énergie issue du freinage qu'il est théoriquement possible de récupérer.

Il est à noter que les valeurs chiffrées de la figure 5.1 correspondent à des valeurs moyennes constatées pour la ligne de Turin en fonction de la fréquence d'occurrence des carrousels.

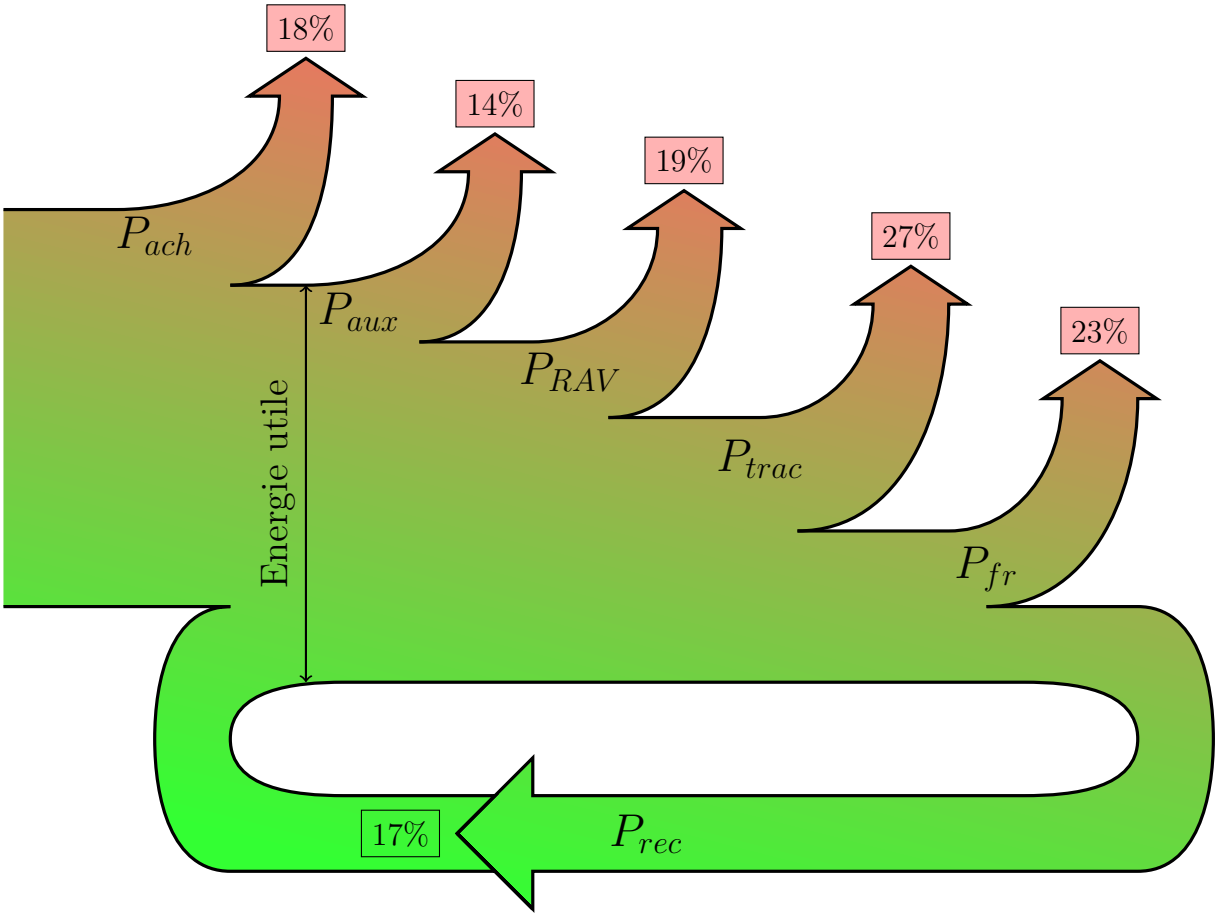


FIGURE 5.1 – Répartition des flux d'énergie dans une ligne de métro automatique comme celle de Turin.

Les pertes énergétiques sont de plusieurs types :

Les pertes d'acheminement P_{ach} : pertes dans les sous-stations et pour la distribution de l'énergie sur la ligne de métro (sec. 2.2.3) ;

Les pertes auxiliaires P_{aux} : pertes liées au fonctionnement des systèmes auxiliaires fixes et embarqués (sec. 2.3.6) ;

Les pertes de résistance à l'avancement P_{RAV} : il s'agit de l'énergie dépensée pour vaincre les forces de frottement mécanique et aérodynamique. Le calcul de ces pertes découle naturellement de l'application du PFD (sec. 2.4.2) ;

Les pertes de la chaine de traction P_{trac} : ce sont les pertes liées à la conversion de l'énergie électrique issue du réseau de traction en énergie mécanique nécessaire à la traction (et inversement) (sec. 2.3.2) ;

Les pertes par freinage P_{fr} : il s'agit de l'énergie cinétique qui n'a pas pu être restituée sur le réseau et a dû être dissipée par dissipation mécanique ou rhéostatique (sec. 2.4.3.2) ;

Le flux P_{rec} correspond à la quantité d'énergie récupérée lors des phases de freinage électrique.

L'énergie utile correspond à la quantité d'énergie nécessaire à la traction. Il s'agit de l'énergie qui doit être injectée sur la ligne électrique pour assurer l'avancement des trains.

Les pertes P_{fr} sont celles que l'on cherche à minimiser dans ces travaux de thèse, en synchronisant les phases de freinage et d'accélération des trains pour favoriser la réutilisation de l'énergie issue du freinage. Ainsi, en moyenne 23% de l'énergie cinétique de traction peut être théoriquement récupérée.

Au chapitre précédent, l'optimisation hors-ligne des temps de stationnement à l'aide de la méthode d'optimisation hybride OEP-AG, pour une journée d'exploitation, a montré qu'en explorant un très faible nombre de solutions possibles, un gain énergétique de 7.8% a été enregistré.

Dans le cadre de l'optimisation temps réel, l'étude des performances de la méthode dans des conditions d'exploitation intégrant des aléas, a été réalisée sur un tour de boucle avec un carrousel établi de 16 trains. La méthode DNFQ a permis d'enregistrer des gains énergétiques de 10,3% et 7,5% pour respectivement un cas d'exploitation avec peu d'aléas et un cas d'exploitation avec un plus grand nombre de perturbations de trafic.

Cependant, ces résultats sont à contraster par rapport à la qualité de la base de données. En effet, cette base de données est composée d'un grand nombre de solutions dont le gain maximal est d'environ 12%. Ainsi, en améliorant la qualité des solutions de la base d'apprentissage, il serait possible d'augmenter encore les gains énergétiques potentiels.

5.2.1.3 Amélioration de la méthodologie par une approche mathématique

La méthodologie développée ici se base sur une optimisation des tables horaires à l'aide de métaheuristiques, dont les solutions sont ensuite apprises pour développer une commande optimale en temps réel.

Certains travaux comme ceux de Peña [36] prennent le parti d'effectuer une résolution mathématique du problème des tables horaires. Cette approche présente l'avantage

de garantir l'optimalité des solutions trouvées, sans toutefois permettre une application en temps réel. La raison principale étant la très grande dimension de l'espace des solutions.

Une méthodologie similaire à celle exposée dans ces travaux pourrait alors être mise en œuvre afin de synthétiser un algorithme de résolution mathématique du problème d'optimisation des tables horaires à l'aide d'une fonction d'approximation. Cela permettrait alors d'assurer une baisse significative des temps de calcul tout en conservant une précision acceptable.

En outre, il est à noter que des approches basées sur la Représentation Energétique Macroscopique (REM) ont été développées dans le but de modéliser l'ensemble des flux énergétiques se produisant sur une ligne de métro automatique [216]. Ces travaux présentent l'intérêt de prendre en compte l'ensemble des éléments énergétiques présent sur la ligne, ce qui permet de calculer assez précisément la consommation énergétique réelle des trains présents sur la ligne.

Il serait alors possible d'améliorer la modélisation et la résolution effectuée au chapitre 2 en utilisant la méthodologie REM évoquée précédemment. Un estimateur neuronal pourrait ensuite être utilisé pour apprendre le comportement de la simulation REM pour en réduire le temps de calcul.

5.2.2 Solutions complémentaires

Ces travaux de thèse se sont focalisés sur la conception d'une méthode logicielle permettant d'améliorer l'efficacité énergétique de lignes ferroviaires. Cette solution présente l'avantage de ne pas nécessiter d'investissement lourd.

D'autres solutions existent pour améliorer encore l'efficacité énergétique des lignes ferroviaires comme l'installation de systèmes de stockage fixes ou embarqués ou l'installation de sous-stations réversibles.

5.2.2.1 Stockage énergétique fixe ou embarqué

Le stockage énergétique stationnaire consiste en une ou plusieurs unités de systèmes de stockage placés le long de la ligne, dont le but est de récupérer l'énergie électrique issue du freinage générée en excès sur la ligne [32].

Le stockage énergétique embarqué consiste quant à lui à équiper les trains en exploitation de systèmes de stockage. Cela permet aux trains de stocker l'énergie issue du freinage qui n'a pas pu être consommée par les trains présents sur la ligne.

Cette dernière solution présente l'intérêt majeur de ne pas impliquer de pertes en lignes supplémentaires liées au renvoi de l'énergie du freinage entre les trains, mais également d'assurer une certaine autonomie énergétique des trains.

5.2.2.2 Sous-stations réversibles

Dans cette étude, nous avons fait l'hypothèse que toutes les sous-stations d'une ligne ferroviaire sont non-réversibles, c'est à dire que l'énergie ne peut transiter que du réseau électrique vers les charges de la ligne.

L'installation de sous-stations réversibles permet ainsi à l'énergie électrique générée en excès d'être renvoyée sur le réseau électrique d'alimentation de la ligne, pour alimenter d'autres clients connectés au réseau.

5.2.2.3 Comparatif de ces solutions

Le tableau 5.1 présente une comparaison des trois solutions précédentes pour quelques critères [217].

La question du coût de ces solutions est un sujet délicat puisque jusqu'au début des années 2010 relativement peu de lignes ferroviaires ont implanté ces solutions. De fait, les exploitants ont une connaissance restreinte de la durée de vie et du retour sur investissement de ces systèmes, ce qui rend difficiles les décisions d'investissement. Parallèlement, ces solutions ne bénéficient pas pour l'instant d'un effet d'expansion de marché qui pourrait faire chuter leur coût.

	Stockage fixe	**Stockage embarqué**	**Sous-stations réversibles**
Aucune modification du matériel roulant	✓		✓
Exploitation sans alimentation électrique (caténaire ou rail d'alimentation)		✓	
Réduction des pertes en ligne		✓	
Baisse des coûts de maintenance du frein mécanique	✓	✓	✓
Limitation des contraintes de sécurité pour les usagers	✓		✓
Lissage de la puissance électrique consommée	✓	✓	
Stabilisation de la tension de la ligne	✓	✓	
Utilisation de l'énergie du freinage pour alimenter d'autres applications			✓
Réduction du nombre de sous-stations	✓		

TABLEAU 5.1 – Comparatif des trois solutions pour différents critères.

Néanmoins, les projets et prototypes se multiplient, et les lignes qui ont intégré ces solutions ont constaté des gains énergétiques significatifs [217], [218].

Les lignes de métro de Rennes et Hanovre ont installé un système de stockage inertiel d'une puissance nominale de 1MW pour une capacité énergétique de 5kWh. L'exploitant de Rennes a effectué une double optimisation, tout d'abord en effectuant une optimisation des tables horaires, puis en mettant en place un volant inertiel au milieu de la ligne. Initialement, la consommation énergétique annuelle de la ligne de Rennes était de 6,5 GWh ; l'optimisation des tables horaires et le système de stockage ont permis d'effectuer des gains énergétique annuels de respectivement 600 MWh et 230 MWh, soit un gain cumulé de 12,7% de la consommation annuelle initiale.

De son côté, l'exploitant de Hanovre a quant à lui enregistré une diminution de sa consommation énergétique annuelle de 462 MWh, soit un gain de 40 k€ en considérant les tarifs d'électricité en Allemagne en 2004.

A Cologne, une solution fixe à base de supercapacités a été privilégiée, pour un gain annuel de 320 MWH.

A Nice, l'installation de batteries $Ni - Mh$ (nickel - hydrure métallique) sur le toit des tramways permet aux trains de se passer d'alimentation électrique sur 11% de la ligne. L'intérêt majeur de cette solution est de préserver le caractère historique du centre ville.

A Manheim, une supercapacité de 1KWh placée sur un tramway a permis de démontrer une baisse de 20% de la consommation globale de ce train.

Une solution embarquée hybride supercapacité-batterie Ni-Mh testée sur le métro de Lisbonne a également engendré un gain énergétique d'environ 11%.

Sur la ligne à haute vitesse reliant Valence (Espagne) à Madrid, cinq sous-stations réversibles ont été implantées, ce qui a permis d'enregistrer une baisse des coûts d'exploitation de 8%.

Tous ces exemples d'applications montrent que des gains énergétiques assez importants peuvent être réalisés en intégrant ces solutions à des lignes en exploitation. Cependant, comme le montre le cas de la ligne de Rennes, une optimisation préalable des tables horaires semble rendre moins attractive les solutions de stockage ou de sous-stations réversibles. Néanmoins, la hausse probable des coûts de l'énergie dans les années à venir devrait garantir une viabilité à long terme, bien que selon les études, son évolution soit assez variable.

Dans une étude réalisée en 2007, l'administration américaine de l'énergie (US EIA) prévoit une stabilisation du coût de l'électricité entre 2005 et 2030 [219], l'Union Européenne prévoit une hausse de 80% du coût moyen de l'électricité entre 2009 et 2030 [220], tandis qu'en France, la tendance actuelle d'évolution des coûts de l'électricité semble indiquer une hausse de 50% entre 2011 et 2030 [221], [222].

5.2.2.4 Insertion de phases de marche sur l'erre

L'utilisation de la marche sur l'erre est également compatible avec la méthodologie développée durant cette thèse. La seule modification à apporter étant de modifier les profils de vitesse commerciale utilisée par les trains lors de la modélisation de l'exploitation de la ligne.

La marche sur l'erre est un mode d'exploitation plutôt utilisé en heure creuse puisque son utilisation diminue la qualité de service de la ligne : le temps de parcours nominal des trains en interstation est volontairement augmenté de quelques secondes en adoptant une conduite plus souple exploitant la déclivité de la ligne. En outre, en période creuse, il est également possible d'étendre la plage de modification des temps de stationnement pour permettre d'améliorer la synchronisation des trains.

En heure de pointe, une telle modification n'est pas possible puisque les trains doivent suivre un cadencement précis afin d'assurer le transit des usagers.

Bibliographie

[1] Julien Damon. L'urbanisation du monde. *Sciences humaines*, (231) :22–27, 2011.

[2] International Energy Agency. A tale of renewed cities. Technical report, International Energy Agency, 2013.

[3] United Nations. World urbanization prospects : the 2007 revision. *United Nations, Department of Economic and Social Affairs (DESA), Population Division, Population Estimates and Projections Section, New York*, 2008.

[4] Mikael Höök and Xu Tang. Depletion of fossil fuels and anthropogenic climate change—a review. *Energy Policy*, 52 :797–809, 2013.

[5] International Energy Agency. *Key world energy statistics*. International Energy Agency, 2015.

[6] Susan Solomon. *Climate change 2007-the physical science basis : Working group I contribution to the fourth assessment report of the IPCC*, volume 4. Cambridge University Press, 2007.

[7] Rajendra K Pachauri, MR Allen, VR Barros, J Broome, W Cramer, R Christ, JA Church, L Clarke, Q Dahe, P Dasgupta, et al. Climate change 2014 : Synthesis report. contribution of working groups i, ii and iii to the fifth assessment report of the intergovernmental panel on climate change. 2014.

[8] Benoit Lefèvre. Urban transport energy consumption : Determinants and strategies for its reduction.. an analysis of the literature. *SAPI EN. S. Surveys and Perspectives Integrating Environment and Society*, (2.3), 2009.

[9] YV Bocharnikov, AM Tobias, and C Roberts. Reduction of train and net energy consumption using genetic algorithms for trajectory optimisation. 2010.

[10] S Açıkbaş and MT Söylemez. Energy wise driving of a mass transit train. In *5th International Conference on Electrical and Electronics Engineering, Bursa, Turkey*, 2007.

[11] Mathias Kinder. Models for periodic timetabling. *Technische Universität, Berlin*, 2008.

[12] Patrick Hachemane. Évaluation de la capacité de réseaux ferroviaires. 1997.

[13] H Bellaiche. Recherche sur la saturation des lignes ferroviaires (rapport d'étape de la phase 1). rapport technique 2166. Technical report, ESF/317-97/RA, SYSTRA, France, 1997.

[14] F Schneider. Recherche sur la saturation des lignes ferroviaires (rapport d'étude de la phase 2). rapport technique 2166. Technical report, ESF/721-97/RA, SYSTRA, France, 1997.

[15] Shi-Qiang Liu. Modelling and solving train scheduling problems under capacity constraints. 2008.

[16] Tewei Chiang, HaiYen Hau, Hwan Ming Chiang, Su Yun Kob, and Chao Ho Hsieh. Knowledge-based system for railway scheduling. *Data & Knowledge Engineering*, 27(3) :289–312, 1998.

[17] Andrew Higgins, Erhan Kozan, and Luis Ferreira. Optimal scheduling of trains on a single line track. *Transportation Research Part B : Methodological*, 30(2) :147–161, 1996.

[18] Keivan Ghoseiri and Fahimeh Morshedsolouk. Acs-ts : Train scheduling using ant colony system. *Advances in Decision Sciences*, 2006, 2006.

[19] Xuesong Zhou and Ming Zhong. Single-track train timetabling with guaranteed optimality : Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B : Methodological*, 41(3) :320–341, 2007.

[20] Raymond SK Kwan and Paavan Mistry. A co-evolutionary algorithm for train timetabling. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 2142–2148. IEEE, 2003.

[21] I Amit and D Goldfarb. The timetable problem for railways. *Developments in Operations Research*, 2 :379–387, 1971.

[22] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4) :550–581, 1989.

[23] Evangelos Bampas, Georgia Kaouri, Michael Lampis, and Aris Pagourtzis. Periodic metro scheduling. In *ATMOS*, 2006.

[24] JE Cury, FAC Gomide, and MJ Mendes. A methodology for generation of optimal schedules for an underground railway system. *Automatic Control, IEEE Transactions on*, 25(2) :217–222, 1980.

[25] Karl Nachtigall and Stefan Voget. A genetic algorithm approach to periodic railway synchronization. *Computers & Operations Research*, 23(5) :453–463, 1996.

[26] SP Gordon and DG Lehrer. Coordinated train control and energy management control strategies. In *Proceedings of the 1998 ASME/IEEE Joint Railroad Conference, 1998.*, pages 165–176. IEEE, 1998.

[27] B Sansé and P Girard. Instantaneous power peak reduction and train scheduling desynchronization in subway systems. 1994.

[28] CS Chang, YH Phoa, W Wang, and BS Thia. Economy/regularity fuzzy-logic control of dc railway systems using event-driven approach. *IEE Proceedings-Electric Power Applications*, 143(1) :9–17, 1996.

[29] P Firpo and S Savio. Optimized train running curve for electrical energy saving in autotransformer supplied AC railway systems. *International Conference on Electric Railways in a United Europe*, pages 23–27, 1995.

[30] T Albrecht. Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control. *Power Supply, Energy Management and Catenary Problems*, page 3, 2004.

[31] Jiann-Fuh Chen, Ray-Lee Lin, and Yow-Chyi Liu. Optimization of an MRT train schedule : reducing maximum traction power by using genetic algorithms. *IEEE Transactions on Power Systems*, 20(3) :1366–1372, 2005.

[32] Petronela-Valeria Buzila. *Gestion énergétique optimale des installations fixes de traction électrique ferroviaire hybrides*. PhD thesis, Lille 1, 2015.

[33] Kyung Kim, Suk-mun Oh, Moonseob Han, and Basic Rate Peak Power. A mathematical approach for reducing the maximum traction energy : the case of korean mrt trains. 2010.

[34] A Nasri, M Fekri Moghadam, and H Mokhtari. Timetable optimization for maximum usage of regenerative energy of braking in electrical railway systems. In *International Symposium on Power Electronics Electrical Drives Automation and Motion (SPEEDAM 2010)*, pages 1218–1221. IEEE, 2010.

[35] Andrés Ramos, María Teresa Peña, Antonio Fernández, and Paloma Cucala. Mathematical programming approach to underground timetabling problem for maximizing time synchronization. In *XI Congreso de Ingeniería de Organización*, pages 1395–1405, 2007.

[36] Maite Peña-Alcaraz, Antonio Fernández, Asuncion Paloma Cucala, Andres Ramos, and Ramon R Pecharromán. Optimal underground timetable design based on power flow for maximizing the use of regenerative-braking energy. *Proceedings of the Institution of Mechanical Engineers, Part F : Journal of Rail and Rapid Transit*, 226(4) :397–408, 2012.

[37] Xin Yang, Bin Ning, Xiang Li, and Tao Tang. A two-objective timetable optimization model in subway systems. *Intelligent Transportation Systems, IEEE Transactions on*, 15(5) :1913–1921, 2014.

[38] J Xun, X Yang, B Ning, T Tang, and W Wang. Coordinated train control in a fully automatic operation system for reducing energy consumption. *Computers in Railways XIII : Computer System Design and Operation in the Railway and Other Transit Systems*, 127 :3, 2013.

[39] Masafumi Miyatake and Hideyoshi Ko. Numerical analyses of minimum energy operation of multiple trains under dc power feeding circuit. In *Power Electronics and Applications, 2007 European Conference on*, pages 1–10. IEEE, 2007.

[40] Brunilde Sansó and Pierre Girard. Train scheduling desynchronization and power peak optimization in a subway system. In *Railroad Conference, 1995., Proceedings of the 1995 IEEE/ASME Joint*, pages 75–78. IEEE, 1995.

[41] C. J. Goodman, B. Mellitt, and N. B. Rambukwella. Computers in railway operations. chapter CAE for the Electrical Design of Urban Rail Transit Systems, pages 173–193. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[42] CJ Goodman, LK Siu, and TK Ho. A review of simulation models for railway systems. In *International Conference on Developments in Mass Transit Systems 1998*, volume 453. IEEE, 1998.

[43] V Van Breusegem, Guy Campion, and Georges Bastin. Traffic modeling and state feedback control for metro lines. *Automatic Control, IEEE Transactions on*, 36(7) :770–784, 1991.

[44] K Kam Wong and TK Ho. Dwell-time and run-time control for dc mass rapid transit railways. *Electric Power Applications, IET*, 1(6) :956–966, 2007.

[45] M Tulbure and R Both. Models for the AC locomotives regenerative braking. In *IEEE International Conference on Automation Quality and Testing Robotics (AQTR 2012)*, pages 525–530. IEEE, 2012.

[46] Zhongbei Tian, Stuart Hillmansen, Clive Roberts, Paul Weston, Lei Chen, Ning Zhao, Shuai Su, and Tingyu Xin. Modeling and simulation of dc rail traction systems for energy saving. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2354–2359. IEEE, 2014.

[47] Mikael Sjöholm. Benefits of regenerative braking and eco driving for high-speed trains : Energy consumption and brake wear. 2011.

[48] Jorge Valero Rodríguez and Javier Sanz Feito. Calculation of remote effects of stray currents on rail voltages in dc railways systems. *IET Electrical Systems in Transportation*, 3(2) :31–40, 2013.

[49] Rémi Vial. *Vers un dimensionnement optimal structure-commande de système multi-convertisseurs. Application aux réseaux de tramways.* PhD thesis, Université de Grenoble, 2012.

[50] C Mayet, Mohamed Mejri, A Bouscayrol, J Pouget, and Yann Riffonneau. Energetic macroscopic representation and inversion-based control of the traction system of a hybrid locomotive. In *2012 IEEE Vehicle Power and Propulsion Conference*, 2012.

[51] TK Ho, YL Chi, J Wang, KK Leung, LK Siu, and CT Tse. Probabilistic load flow in ac electrified railways. In *Electric Power Applications, IEE Proceedings-*, volume 152, pages 1003–1013. IET, 2005.

[52] E Pilo, L Rouco, Alicia Fernandez, and A Hernández-Velilla. A simulation tool for the design of the electrical supply system of high-speed railway lines. In *Power Engineering Society Summer Meeting, 2000. IEEE*, volume 2, pages 1053–1058. IEEE, 2000.

[53] Bih-Yuan Ku and Jen-Sen Liu. Solution of dc power flow for nongrounded traction systems using chain-rule reduction of ladder circuit jacobian matrices. In *Railroad Conference, 2002 ASME/IEEE Joint*, pages 123–130. IEEE, 2002.

[54] Olivier Bossi. Vers la conception optimale d'une electrification ferroviaire. *La revue 3EI*, (74) :9–17, 2013.

[55] CJ Goodman and LK Sin. Dc railway power network solutions by diakoptics. In *Railroad Conference, 1994., Proceedings of the 1994 ASME/IEEE Joint (in Conjunction with Area 1994 Annual Technical Conference)*, pages 103–110. IEEE, 1994.

[56] M Chymera, AC Renfrew, and M Barnes. Analysis of power quality in a dc tram system. 2006.

[57] J. B. Ward and H. W. Hale. Digital computer solution of power-flow problems. *Transactions of the American Institute of Electrical Engineers. Part III : Power Apparatus and Systems*, 75(3-III) :398 – 404, 1956.

[58] Albert M Sasson and Fernando J Jaimes. Digital methods applied to power flow studies. *Power Apparatus and Systems, IEEE Transactions on*, (7) :860–867, 1967.

[59] Xi-Fan Wang, Yonghua Song, and Malcolm Irving. *Modern power systems analysis*. Springer Science & Business Media, 2010.

[60] William F Tinney and Clifford E Hart. Power flow solution by newton's method. *Power Apparatus and Systems, IEEE Transactions on*, (11) :1449–1460, 1967.

[61] Reijer Idema, Domenico Lahaye, Kees Vuik, and Lou Van der Sluis. Fast newton load flow. In *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pages 1–7. IEEE, 2010.

[62] Wei Liu, Qunzhan Li, and Minwu Chen. Study of the simulation of dc traction power supply system based on ac/dc unified newton-raphson method. In *Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on*, pages 1–4. IEEE, 2009.

[63] Yii-Shen Tzeng, Ruay-Nan Wu, and Nanming Chen. Electric network solutions of dc transit systems with inverting substations. *Vehicular Technology, IEEE Transactions on*, 47(4) :1405–1412, 1998.

[64] Y Cai, MR Irving, and SH Case. Iterative techniques for the solution of complex DC-rail-traction systems including regenerative braking. *IEE Proceedings-Generation, Transmission and Distribution*, 142(5) :445–452, 1995.

[65] Seiji Yasunobu, Shoji Miyamoto, and Hirokazu Ihara. A fuzzy control for train automatic stop control. *Trans. of the society of instrument and control engineers*, 2(1) :1–9, 2002.

[66] G Acampora, C Landi, M Luiso, and N Pasquino. Optimization of energy consumption in a railway traction system. In *International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2006. SPEEDAM 2006.*, pages 1121–1126. IEEE, 2006.

[67] YV Bocharnikov, AM Tobias, C Roberts, S Hillmansen, and CJ Goodman. Optimal driving strategy for traction energy saving on dc suburban railways. *Electric Power Applications, IET*, 1(5) :675–682, 2007.

[68] KK Wong and TK Ho. Dynamic coast control of train movement with genetic algorithm. *International journal of systems science*, 35(13-14) :835–846, 2004.

[69] CS Chang and SS Sim. Optimising train movements through coast control using genetic algorithms. In *Electric Power Applications, IEE Proceedings-*, volume 144, pages 65–73. IET, 1997.

[70] S Açıkbaş and MT Söylemez. Coasting point optimisation for mass rail transit lines using artificial neural networks and genetic algorithms. *IET Electric Power Applications*, 2(3) :172–182, 2008.

[71] KK Wong and TK Ho. Coast control of train movement with genetic algorithm. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 2, pages 1280–1287. IEEE, 2003.

[72] KK Wong and TK Ho. Coast control for mass rapid transit railways with searching methods. *IEE Proceedings-Electric Power Applications*, 151(3) :365–376, 2004.

[73] CS Chang and BS Thia. Online rescheduling of mass rapid transit systems : fuzzy expert system approach. In *Electric Power Applications, IEE Proceedings-*, volume 143, pages 307–316. IET, 1996.

[74] KM Kim, KT Kim, and MS Han. A model and approaches for synchronized energy saving in timetabling. *Korea Railroad Research Institute, http ://www. railwayresearch. org/IMG/pdf/a4_kim_kyungmin. pdf*, 2011.

[75] D. Fournier. *Metro regenerative braking optimization through rescheduling : Mathematical Model and Greedy Heuristics Compared to MILP and CMA-ES*. PhD thesis, Université Paris Diderot, 2013.

[76] Siemens. Document technique interne. Technical report, 2007.

[77] David Fournier, Denis Mulard, and François Fages. Optimisation énergétique de tables horaires de métros : une approche hybride. In *JFPC 2012-Huitièmes Journées Francophones de Programmation par Contraintes*, 2012.

[78] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Modeling and solving the train timetabling problem. *Operations research*, 50(5) :851–861, 2002.

[79] P Tormos, A Lova, Federico Barber, L Ingolotti, Montserrat Abril, and Miguel A Salido. A genetic algorithm for railway scheduling problems. In *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, pages 255–276. Springer, 2008.

[80] DJ Epstein, Q Lu, J Zhao, and RC Leachman. An exact solution procedure for determining the optimal dispatching times for complex rail networks. Technical report, Technical report. Los Angeles, CA, USA : Department of Industrial and Systems Engineering, University of Southern California, 2005.

[81] Yann Collette and Patrick Siarry. *Optimisation multiobjectif*. Editions Eyrolles, 2002.

[82] Antonin Ponsich. Stratégies d'optimisation mixte en génie des procédés– application à la conception d'ateliers discontinus. 2005.

[83] Jin-Kao Hao, Philippe Galinier, and Michel Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'intelligence artificielle*, 13(2) :283–324, 1999.

[84] David H Wolpert and William G Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1) :67–82, 1997.

[85] Vincent Gardeux. *Conception d'heuristiques d'optimisation pour les problèmes de grande dimension. Application à l'analyse de données de puces à ADN.* PhD thesis, Université de Paris-Est, 2012.

[86] Nadia Smairi. *Optimisation par essaim particulaire : adaptation de tribes à l'optimisation multiobjectif.* PhD thesis, Université Paris-Est ; École Nationale des Sciences de l'Informatique (Tunis), 2013.

[87] Abbas El Dor. *Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en électronique.* PhD thesis, Université Paris-Est, 2012.

[88] Ahmed Nasreddine Benaichouche. *Conception de métaheuristiques d'optimisation pour la segmentation d'images : application aux images IRM du cerveau et aux images de tomographie par émission de positons.* PhD thesis, Université Paris-Est, 2014.

[89] Charles Darwin. De l'origine des espèces par voie de sélection naturelle. *On the Origin of Species by Means of Natural Selection*, 1859.

[90] Charles Darwin and Clémence-Auguste Royer. *De l'origine des especes ou des lois du progres chez les etres organises par Ch. Darwin.* Guillaumin, 1862.

[91] Ingo Rechenberg. Evolution strategy : Nature's way of optimization. In *Optimization : Methods and applications, possibilities and limitations*, pages 106–126. Springer, 1989.

[92] Thomas Back, David B Fogel, and Zbigniew Michalewicz. *Handbook of evolutionary computation.* IOP Publishing Ltd., 1997.

[93] John H Holland. Adaptation in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence. 1975.

[94] John R Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *IJCAI*, pages 768–774. Citeseer, 1989.

[95] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4) :341–359, 1997.

[96] DT Pham, A Ghanbarzadeh, E Koc, S Otri, S Rahim, and M Zaidi. The bees algorithm–a novel tool for complex optimisation. In *Intelligent Production Machines and Systems-2nd I* PROMS Virtual International Conference 3-14 July 2006*, page 454. Elsevier, 2011.

[97] Craig W Reynolds. Flocks, herds and schools : A distributed behavioral model. In *ACM Siggraph Computer Graphics*, volume 21, pages 25–34. ACM, 1987.

[98] Frank Heppner and Ulf Grenander. A stochastic nonlinear model for coordinated bird flocks. *AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE, WASHINGTON, DC(USA). 1990.*, 1990.

[99] J Kennedy and RC Eberhart. particle swarm optimization, proceedings of ieee international conference on neural networks (icnn'95), 1995.

[100] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1) :58–73, 2002.

[101] James Kennedy, James F Kennedy, Russell C Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.

[102] Yuhui Shi and Russell C Eberhart. Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE, 1999.

[103] Charlie Vanaret. Hybridation d'algorithmes évolutionnaires et de méthodes d'intervalles pour l'optimisation de problèmes difficiles. 2015.

[104] K Premalatha and AM Natarajan. Hybrid pso and ga for global maximization. *Int. J. Open Problems Compt. Math*, 2(4) :597–608, 2009.

[105] Hanaa Hachimi. *Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications*. PhD thesis, INSA de Rouen ; École Mohammadia d'ingénieurs (Rabat, Maroc), 2013.

[106] Ahmed AA Esmin and Stan Matwin. Hpsom : a hybrid particle swarm optimization algorithm with genetic mutation. *International Journal of Innovative Computing, Information and Control*, 9(5) :1919–1934, 2013.

[107] Yi-Tung Kao and Erwie Zahara. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8(2) :849–857, 2008.

[108] Joseph C Pemberton. k-best : A new method for real-time decision making. In *IJCAI*, pages 227–235. Citeseer, 1995.

[109] Francesco Corman, Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C : Emerging Technologies*, 20(1) :79–94, 2012.

[110] Mohammad Ali Shafia, Mohsen Pourseyed Aghaee, Seyed Jafar Sadjadi, and Amin Jamili. Robust train timetabling problem : Mathematical model and branch and bound algorithm. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1) :307–317, 2012.

[111] Twan Dollevoet, Francesco Corman, Andrea D'Ariano, and Dennis Huisman. An iterative optimization framework for delay management and train scheduling. *Flexible Services and Manufacturing Journal*, 26(4) :490–515, 2014.

[112] Johanna Törnquist Krasemann. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C : Emerging Technologies*, 20(1) :62–78, 2012.

[113] Xu Jun Eberlein, Nigel HM Wilson, and David Bernstein. The holding problem with real–time information available. *Transportation science*, 35(1) :1–18, 2001.

[114] Paolo Delle Site and Francesco Filippi. Service optimization for bus corridors with short-turn strategies and variable vehicle size. *Transportation Research Part A : Policy and Practice*, 32(1) :19–38, 1998.

[115] Yuqing Ding and Steven Chien. Improving transit service quality and headway regularity with real-time control. *Transportation Research Record : Journal of the Transportation Research Board*, (1760) :161–170, 2001.

[116] Bingxu Ning, Jing Xun, Smith Gao, and Leiqi Zhang. An integrated control model for headway regulation and energy saving in urban rail transit.

[117] Wei-Song Lin and Jih-Wen Sheu. Adaptive critic design of automatic train regulation of mrt system. In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, pages 1–7. IEEE, 2008.

[118] WS Lin and JW Sheu. Automatic train regulation for metro lines using dual heuristic dynamic programming. *Proceedings of the Institution of Mechanical Engineers, Part F : Journal of Rail and Rapid Transit*, 224(1) :15–23, 2010.

[119] Wei-Song Lin and Jih-Wen Sheu. Metro traffic regulation by adaptive optimal control. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4) :1064–1073, 2011.

[120] Wei-Song Lin and Jih-Wen Sheu. Optimization of train regulation and energy usage of metro lines using an adaptive-optimal-control algorithm. *Automation Science and Engineering, IEEE Transactions on*, 8(4) :855–864, 2011.

[121] Jianwei Qu, Xiaoyun Feng, and Qingyuan Wang. Real-time trajectory planning for rail transit train considering regenerative energy. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2738–2742. IEEE, 2014.

[122] Thomas Albrecht, Andreas Binder, and Christian Gassel. Applications of real-time speed control in rail-bound public transportation systems. *Intelligent Transport Systems, IET*, 7(3) :305–314, 2013.

[123] Pengling Wang, Rob MP Goverde, and Lei Ma. A multiple-phase train trajectory optimization method under real-time rail traffic management. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 771–776. IEEE, 2015.

[124] Jiateng Yin, Dewang Chen, and Lingxi Li. Intelligent train operation algorithms for subway by expert system and reinforcement learning. *Intelligent Transportation Systems, IEEE Transactions on*, 15(6) :2561–2571, 2014.

[125] Jiateng Yin, Dewang Chen, Wentian Zhao, and Long Chen. Online adjusting subway timetable by q-learning to save energy consumption in uncertain passenger demand. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2743–2748. IEEE, 2014.

[126] Yihui Wang, Bart De Schutter, Ton JJ van den Boom, Bin Ning, and Tao Tang. Real-time scheduling for trains in urban rail transit systems using nonlinear optimization. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1334–1339. IEEE, 2013.

[127] Yihui Wang, Bin Ning, Tao Tang, Ton JJ van den Boom, and Bart De Schutter. Efficient real-time train scheduling for urban rail transit systems using iterative convex programming. *Intelligent Transportation Systems, IEEE Transactions on*, 16(6) :3337–3352, 2015.

[128] Nicolas Balacheff. Didactique et intelligence artificielle. *Recherches en didactique des mathématiques (Revue)*, 14 :9–42, 1994.

[129] Joao Cunha, Rui Serra, Nuno Lau, Luís Seabra Lopes, and Antóio JR Neves. Batch reinforcement learning for robotic soccer using the q-batch update-rule. *Journal of Intelligent & Robotic Systems*, pages 1–15, 2015.

[130] Steve Dini and Mark Serrano. Combining q-learning with artificial neural networks in an adaptive light seeking robot. 2012.

[131] Guoqiang Peter Zhang. Neural networks for classification : a survey. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 30(4) :451–462, 2000.

[132] Anchal Garg and Rohit Bajaj. Facial expression recognition & classification using hybridization of ica, ga, and neural network for human-computer interaction. *Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org*, 2(1), 2015.

[133] Fred Richardson, Douglas Reynolds, and Najim Dehak. A unified deep neural network for speaker and language recognition. *arXiv preprint arXiv :1504.00923*, 2015.

[134] Jibendu Kumar Mantri, P Gahan, and Braja B Nayak. Artificial neural networks–an application to stock market volatility. *Soft-Computing in Capital Market : Research and Methods of Computational Finance for Measuring Risk of Financial Instruments*, page 179, 2014.

[135] Iebeling Kaastra and Milton Boyd. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3) :215–236, 1996.

[136] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making : The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2) :427–436, 2008.

[137] Harry B Burke. Artificial neural networks for cancer research : outcome prediction. In *Seminars in Surgical Oncology*, volume 10, pages 73–79. Wiley Online Library, 1994.

[138] Amirhosein Shantia, Eric Begue, and Marco Wiering. Connectionist reinforcement learning for intelligent unit micro management in starcraft. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1794–1801. IEEE, 2011.

[139] Jacob Schrum and Risto Miikkulainen. Evolving multimodal behavior with modular neural networks in ms. pac-man. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 325–332. ACM, 2014.

[140] Xiao Li. Intelligent learning control of hydraulic flow regulating pump with neural network load flow identifier. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 2, pages 539–543. IEEE, 2009.

[141] Adrian K Agogino and Kagan Tumer. Quicker q-learning in multi-agent systems. *Online], www. citeseerx. ist. psu. edu/viewdoc/download*.

[142] Rémi Coulom. *Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2002.

[143] S Kamalasadan, AK Srivastava, and D Thukaram. Novel algorithm for online voltage stability assessment based on feed forward neural network. In *Power Engineering Society General Meeting, 2006. IEEE*, pages 7–pp. IEEE, 2006.

[144] TT Nguyen. Neural network load-flow. *IEE Proceedings-generation, transmission and distribution*, 142(1) :51–58, 1995.

[145] D Thukaram, K Harish Kashyap, et al. Artificial neural network application to power system voltage stability improvement. In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, volume 1, pages 53–57. IEEE, 2003.

[146] E-D Aparaschivei, Ovidiu Ivanov, and Mihai Gavrilas. Load flow estimaton in electrical systems using artificial neural networks. In *Electrical and Power Engineering (EPE), 2012 International Conference and Exposition on*, pages 276–279. IEEE, 2012.

[147] S Jemeı, Daniel Hissel, Marie-Cécile Péra, and Jean-Marie Kauffmann. On-board fuel cell power supply modeling on the basis of neural network methodology. *Journal of Power Sources*, 124(2) :479–486, 2003.

[148] De la médecine : la mémoire. http://cepheides.fr/article-de-la-medecine-la-memoire-125500338.html. Accessed : 2015-11-17.

[149] IA Basheer and M Hajmeer. Artificial neural networks : fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1) :3–31, 2000.

[150] Nishesh Kumar, Rohit Wangneo, PK Kalra, and SC Srivastava. Application of artificial neural networks to load flow solutions. In *TENCON'91.1991 IEEE Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems*, volume 1, pages 199–203. IEEE, 1991.

[151] WL Chan, ATP So, and LL Lai. Initial applications of complex artificial neural networks to load-flow analysis. *IEE Proceedings-Generation, Transmission and Distribution*, 147(6) :361–366, 2000.

[152] Amit Jain, SC Tripathy, R Balasubramanian, and Yoshiyuki Kawazoe. Stochastic load flow analysis using artificial neural networks. In *IEEE Power Engineering Society General Meeting, 2006.*, pages 6–14. IEEE, 2006.

[153] A Rathinam, S Padmini, and V Ravikumar. Application of supervised learning artificial neural networks [cpnn, bpnn] for solving power flow problem. 2007.

[154] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks, 1989.*, pages 593–605. IEEE, 1989.

[155] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks : Tricks of the trade*, pages 9–48. Springer, 2012.

[156] Marc Mezard and Jean-P Nadal. Learning in feedforward layered networks : The tiling algorithm. *Journal of Physics A : Mathematical and General*, 22(12) :2191, 1989.

[157] Mario Marchand, Mostefa Golea, and Pál Ruján. A convergence theorem for sequential learning in two-layer perceptrons. *EPL (Europhysics Letters)*, 11(6) :487, 1990.

[158] Scott E Fahlman and Christian Lebiere. The cascade-correlation learning architecture. 1989.

[159] D Martinez and D Esteve. The offset algorithm : Building and learning method for multilayer neural networks. *EPL (Europhysics Letters)*, 18(2) :95, 1992.

[160] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[161] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9) :1481–1497, 1990.

[162] Jan Depenau. Automated design of neural network architecture for classification. *DAIMI Report Series*, 24(500), 1995.

[163] Babak Hassibi and David G Stork. *Second order derivatives for network pruning : Optimal brain surgeon*. Morgan Kaufmann, 1993.

[164] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPs*, volume 89, 1989.

[165] Yann Lecun, Patrice Y. Simard, and Barak Pearlmutter. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. In *Advances in Neural Information Processing Systems*, pages 156–163. Morgan Kaufmann, 1993.

[166] J Sola and J Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3) :1464–1468, 1997.

[167] S Saarinen, R Bramley, and G Cybenko. Ill-conditioning in neural network training problems. *SIAM Journal on Scientific Computing*, 14(3) :693–714, 1993.

[168] Chanabasayya Vastrad et al. Performance analysis of neural network models for oxazolines and oxazoles derivatives descriptor dataset. *arXiv preprint arXiv :1312.2853*, 2013.

[169] Aymen Cherif. *Réseaux de neurones, SVM et approches locales pour la prévision de séries temporelles*. PhD thesis, Tours, 2013.

[170] D Randall Wilson and Tony R Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10) :1429–1451, 2003.

[171] Thomas Czernichow. Architecture selection through statistical sensitivity analysis. In *Artificial Neural Networks—ICANN 96*, pages 179–184. Springer, 1996.

[172] Mark B Milam. *Real-time optimal trajectory generation for constrained dynamical systems*. PhD thesis, California Institute of Technology, 2003.

[173] Malachy Carey. Optimizing scheduled times, allowing for behavioural response. *Transportation Research Part B : Methodological*, 32(5) :329–342, 1998.

[174] Rob MP Goverde. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C : Emerging Technologies*, 18(3) :269–287, 2010.

[175] Xuesong Zhou and Muhammad Babar Khan. Slack time allocation in robust double-track train timetabling applications. In *Transportation Research Board 87th Annual Meeting*, number 08-2971, 2008.

[176] Michiel JCM Vromans, Rommert Dekker, and Leo G Kroon. Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2) :647–665, 2006.

[177] Francesco Corman and Limin Meng. A review of online dynamic models and algorithms for railway traffic management.

[178] Marc Blancheteau. *L'apprentissage chez l'animal : faits et théories*, volume 114. Editions Mardaga, 1979.

[179] Kent C Berridge and Terry E Robinson. What is the role of dopamine in reward : hedonic impact, reward learning, or incentive salience ? *Brain Research Reviews*, 28(3) :309–369, 1998.

[180] Richard Bellman. A markovian decision process. Technical report, DTIC Document, 1957.

[181] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[182] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1) :9–44, 1988.

[183] Emmanuel Rachelson. *Problèmes décisionnels de Markov temporels : formalisation et résolution*. PhD thesis, 2009.

[184] Sebastian B Thrun. The role of exploration in learning control. *Handbook of intelligent control : Neural, fuzzy and adaptive approaches*, 1992.

[185] Steffen Nissen. Large scale reinforcement learning using q-sarsa (λ) and cascading neural networks. *Unpublished masters thesis, Department of Computer Science, University of Copenhagen, København, Denmark*, 2007.

[186] Steve Dini and Mark Serrano. Combining q-learning with artificial neural networks in an adaptive light seeking robot. 2012.

[187] Marco A Wiering. Explorations in efficient reinforcement learning. 1999.

[188] Rémi Munos. Programmation dynamique avec approximation de la fonction valeur. *Processus décisionnels de Markov et intelligence artificielle*, pages 19–50, 2008.

[189] Marc Hayoun. La méthode de monte carlo metropolis. *École «Simulation Numérique en Matière Condensée»(Paris, 29-31 mai 2002)*, 2002.

[190] Richard S Sutton and Andrew G Barto. *Reinforcement learning : An introduction*, volume 1. MIT press Cambridge, 1998.

[191] Dmitry Kamenetsky. *A comparison of neural network architectures in reinforcement learning in the game of othello*. PhD thesis, University of Tasmania, 2005.

[192] Antoine Cornuéjols and Laurent Miclet. *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles, 2011.

[193] Satinder P Singh and Richard S Sutton. Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1-3) :123–158, 1996.

[194] Peter Vamplew and Robert Ollington. Global versus local constructive function approximation for on-line reinforcement learning. In *AI 2005 : Advances in Artificial Intelligence*, pages 113–122. Springer, 2005.

[195] David Garcıa and Leonardo Garrido. Generation of motion policies applying multiagent reinforcement learning in simulated robotic soccer.

[196] Remi Munos and Andrew W Moore. Variable resolution discretization for high-accuracy solutions of optimal control problems. *Robotics Institute*, page 256, 1999.

[197] Stephan Pareigis. Adaptive choice of grid and time in reinforcement learning. In *NIPS*. Citeseer, 1997.

[198] Thomas Degris, Patrick M Pilarski, and Richard S Sutton. Model-free reinforcement learning with continuous action in practice. In *American Control Conference (ACC), 2012*, pages 2177–2182. IEEE, 2012.

[199] Alexander A Sherstov and Peter Stone. Function approximation via tile coding : Automating parameter choice. In *Abstraction, Reformulation and Approximation*, pages 194–205. Springer, 2005.

[200] Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *Machine Learning : ECML 2005*, pages 317–328. Springer, 2005.

[201] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4) :160–163, 1991.

[202] Kao-Shing Hwang and Chia-Yue Lo. Policy improvement by a model-free dyna architecture. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(5) :776–788, 2013.

[203] Takeshi Tateyama, Seiichi Kawata, and Yoshiki Shimomura. Parallel reinforcement learning systems using exploration agents and dyna-q algorithm. In *SICE, 2007 Annual Conference*, pages 2774–2778. IEEE, 2007.

[204] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the seventh international conference on machine learning*, pages 216–224, 1990.

[205] Guillaume Laurent. *Synthèse de comportements par apprentissages par renforcement parallèles : application à la commande d'un micromanipulateur plan.* PhD thesis, Université de Franche-Comté, 2002.

[206] Andrés Pérez-Uribe and Eduardo Sanchez. A comparison of reinforcement learning with eligibility traces and integrated learning, planning and reacting. In *Computational Intelligence for Modelling, Control and Automation*, pages 154–159, 1999.

[207] Sofia Zaidenberg. *Apprentissage par renforcement de modeles de contexte pour l'informatique ambiante.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2009.

[208] Georgi Petkov, Tchavdar Naydenov, Maurice Grinberg, and Boicho Kokinov. Building robots with analogy-based anticipation. In *KI 2006 : Advances in artificial intelligence*, pages 76–90. Springer, 2007.

[209] Martin Riedmiller. 10 steps and some tricks to set up neural reinforcement controllers. In *Neural Networks : Tricks of the Trade*, pages 735–757. Springer, 2012.

[210] Zoran Miljković, Marko Mitić, Mihailo Lazarević, and Bojan Babić. Neural network reinforcement learning for visual control of robot manipulators. *Expert Systems with Applications*, 40(5) :1721–1736, 2013.

[211] Philippe Cheynet. *Etude de la robustesse du contrôle intelligent face aux fautes induites par les radiations.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1999.

[212] MD Emmerson and RI Damper. Relations between fault tolerance and internal representations for multi-layer perceptrons. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 2, pages 281–284. IEEE, 1992.

[213] Kiyotoshi Matsuoka. Noise injection into inputs in back-propagation learning. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3) :436–440, 1992.

[214] A González-Gil, R Palacin, P Batty, and JP Powell. A systems approach to reduce urban rail energy consumption. *Energy Conversion and Management*, 80 :509–524, 2014.

[215] Walter Gunselmann. Technologies for increased energy efficiency in railway systems. In *Power Electronics and Applications, 2005 European Conference on*, pages 10–pp. IEEE, 2005.

[216] Clement MAYET, Philippe Delarue, Alain Bouscayrol, Eric CHATTOT, and Jean Noel VERHILLE. Comparison of different emr-based models of traction power substations for energetic studies of subway lines. 2015.

[217] The "Ticket to Kyoto". Wp2b energy recovery –an overview of braking energy recovery technologies in the public transport field. Technical report, Projet Européen, 2011.

[218] Arturo Gonzalez-Gil, Roberto Palacin, Paul Batty, and Jonathan P Powell. Energy-efficient urban rail systems : strategies for an optimal management of regenerative braking energy. In *Transport Research Arena (TRA) 5th Conference : Transport Solutions from Research to Deployment*, 2014.

[219] Annual Energy Outlook. with projections to 2030. *US Energy Information Administration*, 2007.

[220] Pantelis Capros, Leonidas Mantzos, Nikos Tasios, Alessia De Vita, and Nikolaos Kouvaritakis. *EU Energy Trends to 2030 : Update 2009*. Publications Office of the European Union, 2010.

[221] http://www.statistiques.developpement-durable.gouv.fr/donnees-ligne/r/pegase.html. Accessed : 2016-02-22.

[222] http://www.developpement-durable.gouv.fr/IMG/pdf/11-0362_5A_ET_note_synthese_sc_pros_v3.pdf. Accessed : 2016-02-22.

OPTIMISATION DE LA CONSOMMATION ÉNERGÉTIQUE D'UNE LIGNE DE MÉTRO AUTOMATIQUE PRENANT EN COMPTE LES ALÉAS DE TRAFIC A L'AIDE D'OUTILS D'INTELLIGENCE ARTIFICIELLE

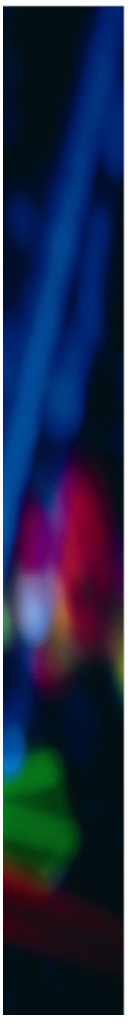
RESUME : En 2014, dans le cadre du Plan Climat, les pays membres de l'Union Européenne, se sont engagés à réduire de près de 27% leur consommation d'énergie. L'un des axes d'études concerne l'augmentation de l'efficacité énergétique des transports urbains. Cette thèse a pour objectif de proposer une méthodologie afin de réduire la consommation énergétique de lignes de métro automatique tout en intégrant les perturbations de trafic qui se produisent dans des conditions normales d'exploitation. Le principe retenu dans ces travaux est de maximiser la réutilisation de l'énergie générée lors du freinage des trains, par les autres trains présents sur la ligne. Une première partie est dédiée à la modélisation électrique d'une ligne de métro automatique et à la présentation de méthodes permettant de calculer les flux de puissances entre les trains et les sous-stations d'alimentation. Ensuite, des algorithmes d'optimisation sont introduits pour effectuer l'optimisation des paramètres d'exploitation les plus influents dans une configuration idéale n'intégrant pas les aléas de trafic. Enfin, une méthodologie basée sur un apprentissage des données de simulation est développée dans le but de réaliser l'optimisation énergétique de la consommation en temps réel et en intégrant les perturbations de trafic. Cette dernière partie aura ainsi pour objectif de fournir une aide à la décision dans le choix des temps d'arrêts que doivent effectuer chaque train en station afin de maximiser la récupération de l'énergie issue du freinage.

Mots clés : Intelligence artificielle, Optimisation dynamique, Métaheuristiques, Synchronisation ferroviaire, Efficacité énergétique, Régulation de trafic

ENERGY CONSUMPTION OPTIMIZATION OF AN AUTOMATIC METRO LINE INTEGRATING TRAFFIC FLUCTUATIONS WITH ARTIFICIAL INTELLIGENCE TOOLS

ABSTRACT : In 2014, as part of the Climate Plan, EU member countries have committed to reduce by 27% their energy consumption. One of the main focal areas consists in increasing the energy efficiency of urban transports. This thesis aims to propose a methodology to reduce the energy consumption of automatic metro lines while integrating traffic disruptions that occur under normal operating conditions. The principle adopted in this work is to maximize the reuse of electrical energy generated during braking of the train, by other trains running on the line. First part is dedicated to the electrical modeling of an automatic metro line and development of methods to calculate power flows between trains and power substations. Then, optimization algorithms are introduced to perform optimization of the most influential operating parameters in an ideal configuration ignoring traffic fluctuations. Finally, a methodology based on learning simulation data is developed in order to achieve optimization of energy consumption integrating traffic disruptions in real time. This last part will thus purchase the objective to provide a decision support to determine optimal dwell times to be carried out by trains in each station, so as to maximize braking energy recovery.

Keywords : Artificial intelligence, Dynamic optimization, Metaheuristics, Railway synchronization, Energy efficiency, Traffic regulation



ARTS ET MÉTIERS ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY